THE ROUTING AND SCHEDULING OF A CLASS OF POSTAL VEHICLES

A THESIS

Presented to

The Faculty of the Division of Graduate

Studies and Research

By

Robert Alan Holmes

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in Industrial and Systems Engineering

Georgia Institute of Technology

January 1975

THE ROUTING AND SCHEDULING OF

A CLASS OF POSTAL VEHICLES

Approved:

Robert G. Parker, Chairman

Mokhtar S. Bazaraa

Richard H. Deane

Date approved by Chairman: 1-28-75

ACKNOWLEDGMENTS

For his helpful guidance and insights, I wish to thank my thesis advisor, Dr. Robert G. Parker. He has been responsible for helping initiate my interest in this problem, and his personal concern has been vital to this work's completion.

The knowledge and experience of both Dr. Mokhtar S. Bazaraa and Dr. Richard H. Deane, the members of my reading committee, have been very valuable.

In addition, Dr. William W. Hines deserves my appreciation for his efforts in all aspects of my graduate work.

Most importantly, I wish to thank my wife, Cindi, for her inspiration both within the academic environment and outside it. Her patience, her understanding, and the constancy of her will joining and aiding mine have been essential. The importance of her contributions cannot be overstated.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

SUMMARY

The general vehicle scheduling problem has been examined in the context of the constraints imposed by the United States Postal Service. When this is done, two important considerations arise: (1) the need to solve problems containing many (at least 100) demand points easily; and (2) the need to treat non-symmetric costs.

The first consideration limits the treatment of this type of problem to heuristic solution procedures. The second consideration requires a new procedure in that all existent techniques apply to symmetric costs.

An algorithm is presented which solves non-symmetric vehicle scheduling problems. This algorithm is shown to be capable of solving problems with 100 demand points without excessive computational time. Computational effort for the non-symmetric problem is no greater than that for the symmetric problem. The effect of different truck capacities upon solution time is shown.

A modification to the basic procedure is also presented. This modification is shown to be capable of improving solutions. Computational results of this modification for both symmetric and non-symmetric problems are given.

CHAPTER I

INTRODUCTION

Routing and scheduling of vehicles have given rise to many problematical situations of interest. This interest has risen concomittantly with the dependence upon motorized transport for shipment of goods. In the 1960's approximately three out of every four tons of goods moved in the United States were moved at least part of the way by trucks [16]. In addition, recent increases associated with transportation costs and concern with energy conservation make efficient utilization of vehicles increasingly desirable.

The general problem of concern can be stated simply. One must service (have goods collected, delivered, or both); a known set of stations (demand points); with a fleet of vehicles of known size. The vehicles are originally located at a central facility, from which all vehicle trips must originate and conclude. The objective is to minimize some measure of system effectiveness; e.g., total distance traveled, total time consumed, total cost incurred, etc. No restriction is placed upon a time or a time interval in which the demands must be serviced. This problem has been alternately referred to as the vehicle scheduling problem, the vehicle routing problem, or the truck dispatching problem. It shall be referred to here as the vehicle scheduling problem.

The utility of such analysis enjoys wide diversity: e.g., military uses involved with transport or deployment of goods, manufacturers' needs to move goods from a production facility to distribution warehouses, the distribution warehouses need to distribute locally, deployment of fork-lifts within a warehouse, and demand-based mass transit systems. In addition, related problems, such as determining the smallest number of vehicles which can meet a given demand situation, efficient use of an airplane fleet to meet daily schedules, garbage collection, taxi fleet utilization, police patrol car deployment, could all benefit from this type of analysis.

The general vehicle scheduling problem shall be treated here in relation to some of the constraints imposed by a system of collection or distribution of mails by United States Postal Service vehicles. These collection and distribution activities range from foot collection of single letters, to transfer of large quantities of mail between main stations. We shall assume that all of these activities can be approximated by demands (collection or distribution) at discrete points. The Postal Service situation can then be formulated as a vehicle scheduling problem.

Transporting mails is an acute problem for the Postal Service. In 1973, transportation costs accounted for 45 percent of all expenses other than personnel within the postal system [1]. Generally, all transportation over one or two hundred miles is done by air [1]. This implies that most ground transport involves small distances. The air transport is typically inter-city travel, and the ground transport is typically intra-city travel. Hence, Postal Service vehicles usually operate in or near metropolitan areas. In intra-city travel, the cost (cost shall be used to represent time, distance, or other system parameter to be optimized) associated with transport from demand point i to demand point j may not be the same as transport from j to i. This nonsymmetry of cost may be caused by traffic congestion, traffic signals, road hazards, and other considerations. This forces some uniqueness of the problem, in that many current solution procedures require these costs to be identical. Not accounting for these non-symmetric costs could induce considerable error. These factors will be discussed further in following sections.

Another facet of the postal system is the size of problems which must be handled. Servicing 100 demand points or more in a large metropolitan area can be expected. Also, due to demographic changes, the demands of points to be serviced and the location and number of points to be serviced may change rapidly. These two factors require fast, easy solutions for large problems. Hence, the constraints imposed upon the general vehicle scheduling problem by the United States Postal System are: (1) non-symmetric costs, (2) large problem size, (3) the necessity for frequent solutions.

Due to the variety of operations performed by the United States Postal Service, some are best described as operations performed along an arc (street) rather than servicing demands at given points. This is a somewhat different problem referred to as the Chinese Postman Problem (see Emonds [15], Beltramie and Bodin [8], and Orloff [35]). Here only the class of transport servicing demand points shall be considered.

This thesis will be concerned with solutions to the general vehicle routing and scheduling problem with constraints as described above by a particular system, the United States Postal Service collection or distribution of mail.

Literature Review

For clarity and simplicity of presentation, the following notation shall be utilized throughout this discussion:

n represents the number of stations or demand points to be serviced inclusive of the central facility.

m represents the total number of vehicles available to service the n demand points.

m' represents the number of vehicles utilized in any feasible solution. This is also the number of routes in the solution.

$d_{i,j}$ represents the system cost parameter to be optimized associated with transport from point i to point j.

D represents the n×n matrix representation of all inter-point costs $d_{i,j}$.

K represents the cost of any feasible solution.

q_i represents the demand to be serviced at point i.

Q_z represents the current demand of the zth route.

C_k represents the capacity of the kth size truck. $C_k = C$ if all trucks are of the same size.

T_k represents the number of trucks available of size k.

$x_{i,j}$ represents an integer variable assigned a value of 0 if the arc i,j is not traversed, and assigned a value of -1 if the arc is traversed by a truck. If an arc is traversed twice, this must be done by the same truck and $x_{i,j}$ then is given a value of -2.

$s_{i,j}$ represents the savings (to be defined later) associated with joining point i and point j on a vehicle route.

S represents the savings matrix of all $s_{i,j}$.

t_z represents the number of demand points assigned to the zth route, $\sum_{z=1}^{m'} t_z = n - 1$

The first comprehensive assessment of the vehicle scheduling problem was presented by Dantzig and Ramser [13]. They viewed the problem as a generalization of the traveling salesman problem (see Bellmore and Malone [7] and Bellmore and Nemhauser [5] for a discussion of the traveling salesman problem) in which n cities are to be visited once and only once by a single salesman, incurring minimum cost. They chose distance as the system variable for improvement and assumed a symmetric ($d_{i,j} = d_{j,i}, \forall_{i,j}$) distance matrix, D. They assumed all trucks were of uniform size, C, and that the truck size was much smaller than the sum of all demands, such that

$$C << \sum_{i=2}^{n} q_i$$

They also assumed that enough trucks exist to initially assign one truck to each demand point. If this is not true, "dummy" assignments can be made, as suggested by Tillman and Cochran [42]. This will not affect the solution as long as a feasible assignment of "real" trucks exists.

Dantzig and Ramser then ranked all demands such that,

$$q_1 < q_2 < \ldots < q_{n-1}$$

and determined the largest t such that

$$\sum_{i=1}^{t} q_i \leq C \text{ (or } \sum_{i=1}^{t+1} q_i > C)$$

They then determined the number of stages, N, of "aggregation," or stages used to form final routes by

$$N \simeq \log_2 t$$

In the gth stage of aggregation they allowed joining of points on a route such that the total demand serviced on that route does not exceed $C/2^{N-g}$. Their object was to minimize total cost,

$$\text{total cost} = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{i,j}\, x_{i,j} \quad , \quad i \neq j$$

Their method of selection for the joining of points did not guarantee optimality, but their techniques have served as a basis for many popular heuristic solution procedures existent today.

Perhaps the most successful adaptation of the above procedures was formulated by Clarke and Wright [10]. They recognized that limiting the filling of trucks at each stage of aggregation might be so restrictive as to detract from the quality of the final solution. They therefore chose to fill all vehicles as near capacity as possible, without regard to steps of aggregation.

Clarke and Wright started with the same initial solution of one truck assigned to each demand point. They then computed a "savings" for each possible joining of points on a route. The savings for joining i to j is determined, with reference to Figure 1 as follows:

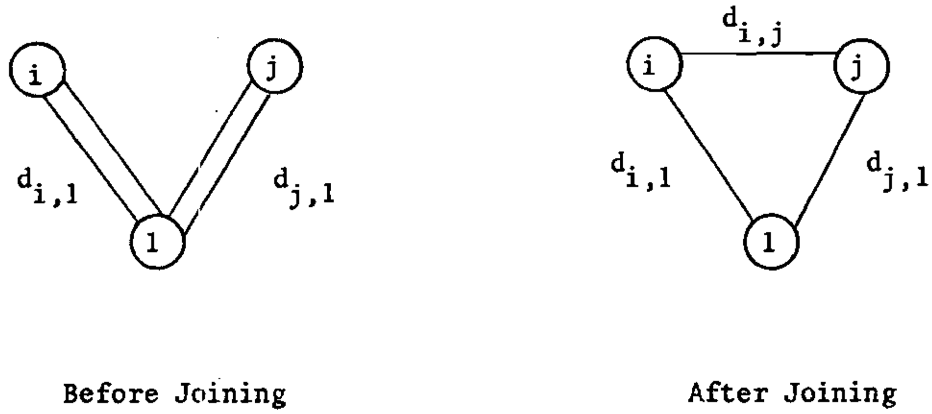


Figure 1. Demand Points Before and After Being Joined on a Route Having Symetric Costs

1. The original cost of i and j not joined is:

   $d_{1,i} + d_{i,1} + d_{1,j} + d_{j,1}$.

2. The cost after joining is:

   $d_{1,i} + d_{j,1} + d_{i,j}$.

3. The "savings" associated with joining i and j is:

   $$s_{i,j} = d_{1,i} + d_{i,1} + d_{1,j} + d_{j,1} - (d_{1,i} + d_{j,1} + d_{i,j})$$
   $$= d_{i,1} + d_{1,j} - d_{i,j}.$$

In this manner all possible savings are calculated. These savings can be tabulated as a matrix. Sample cost and savings matrices are shown in Figure 2. The pair of points having maximum savings, feasible with respect to truck capacities and any other constraints, are then joined. This process is continued until all possible joinings are made. Clarke and Wright also allowed for the consideration of different vehicle sizes. However, the vehicle capacities serve only to act as a check after the pair for joining has been selected, and

	1	2	3	4	... j
1		$d_{1,2}$	$d_{1,3}$	$d_{1,4}$	... $d_{1,j}$
2	$d_{2,1}$		$d_{2,3}$	$d_{2,4}$	... $d_{2,j}$
3	$d_{3,1}$	$d_{3,2}$		$d_{3,4}$	... $d_{3,j}$
4	$d_{4,1}$	$d_{4,2}$	$d_{4,3}$		... $d_{4,j}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
i	$d_{i,1}$	$d_{i,2}$	$d_{i,3}$	$d_{i,4}$	... $d_{i,j}$

Cost Matrix

	1	2	3	4	... j
1					
2			$S_{2,3}$	$S_{2,4}$	... $S_{2,j}$
3		$S_{3,2}$		$S_{3,4}$	... $S_{3,j}$
4		$S_{4,2}$	$S_{4,3}$		... $S_{4,j}$
$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$
i		$S_{i,2}$	$S_{i,3}$	$S_{i,4}$	... $S_{i,j}$

Savings Matrix

Figure 2. Sample Cost and Savings Matrices

the effects of capacity restrictions upon optimality are not considered in the selection process. This procedure will be discussed in Chapter II in greater detail, as it is used as a basis for this work.

Hayes [23] identifies the two major faults with the above method: (1) once a pair is joined it is never reconsidered, (2) matching of demands to capacities is not directly considered. These considerations permit suboptimal solutions. Hayes has presented two examples in which the Clarke and Wright procedure gives a suboptimal solution. Both of these examples are solved in Chapter III.

Tillman and Cochran [42] improved the solution quality of the Clarke and Wright method by using a "look-ahead" procedure. This procedure examines the total savings from all possibilities of two successive joinings. The first of the two pairs of points which maximizes the total savings is then joined. They also present an example in which the Clarke and Wright procedure gives a suboptimal solution. This example is also solved in Chapter III. Continuing this idea, Hering [25] explores the possibilities of an "extended look-ahead procedure," examining three, four, or more successive joinings at one time. But as Tillman and Cochran note, for large numbers of successive joinings, the cost in computation time begins to exceed the value gained from looking ahead. This method approaches total enumeration as the number of successive look-aheads approaches the number of joined point pairs in the final solution.

Tillman [41] extends the Clarke and Wright procedure to include probabilistic demands and multiple terminals (central facilities). Though of interest, neither case will be considered here. Cochran

[11] discusses how Clarke and Wright's method may be adapted to some unusual cases, i.e., individual demands exceeding vehicle capacity. He also suggests a possible improvement involving reassignment of trucks to demand points. Smith [38] investigates the line haul problem which is concerned with demand points, both the shipping and receiving of a commodity. Smith is concerned with the minimization of "back-haul" distance. He uses a savings approach for solution.

Gaskell [19] compared several criteria based on inter-point costs (and bearing some relation to savings) against the savings criteria. While he determined that other measures could be as efficient, he found none superior to savings in providing high quality solutions.

Other heuristic methods bearing no relation to Clarke and Wright's method have appeared. Hayes [23] developed a technique determining the desirability of joining a point based on factors including: demand at the point, distance from the terminal, distance from joined points, and distance from other unjoined points. Also included was a random factor with no relation to any of the characteristics of the point. Points are joined on a basis of this desirability factor until all possible joinings are exhausted and the traveling salesman problem is solved for each route. This procedure is repeated a number of times, keeping the best solution. Hayes notes that the procedure is problem dependent, achieving poorer results when the terminal is located at or near the perimeter of points considered.

Several observations become evident. Solving the traveling salesman problem for each route could become time consuming in larger problems. Realistic problems may have over 100 demand points and the number of points assigned to a route can become fairly large (see results in Chapter III). In addition, Hayes' use of Karg and Thompson's [26] heuristic solution to the traveling salesman problem introduces increased non-optimality. The additive effect of this could be highly significant. There is also a large amount of subjectivity present, e.g., determining the weights to be given to each factor considered, determining what factors to use, and especially the random element.

The role of the random element in Hayes' procedure suggests that perhaps some form of random generation of routes with systematic improvement should also be considered. This was done by Gillette and Miller [21] with some success. They deploy the demand points on a Euclidean coordinate system. These points are joined on routes based on radial angle from an arbitrary reference line until truck capacities are met. The generation of this initial solution is illustrated in Figure 3.

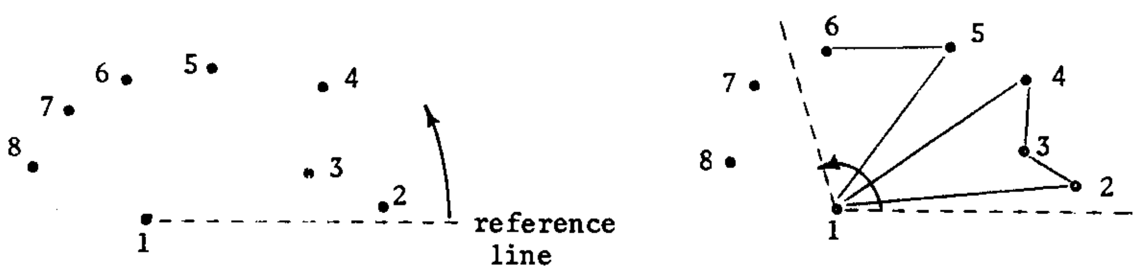


Figure 3. Generation of An Initial Solution by Gillette and Miller

These routes are then improved by a systematic interchange of points. A number of solutions are found in this manner, until a solution of desired quality is obtained.

Still another technique was attempted by Newton and Thomas [33]. They first solved the traveling salesman problem for the same distance matrix, D, and then subdivided this solution into routes. Their results showed the quality of a solution so obtained varied depending on the number of points on a route. Also, the results of Svestka and Huckfeldt [40] show that for their procedure, the one-salesman traveling salesman problem is computationally more difficult than a problem having more than one salesman. The traveling salesman problem with more than one salesman is similar to the vehicle scheduling problem. In view of this, it seems illogical to first solve the traveling salesman problem.

Christofides and Eilon [9] present a three-optimal heuristic method based upon work done by Lin [28] concerning the traveling salesman problem. Their method determines a solution which cannot be improved by replacing any three links (joinings) in the given solution. This method can be extended to four-optimal, five-optimal, etc., but three is selected as a balance between solution quality and computation time. They report, in general, an improvement in solution quality, but at the expense of greater computation time.

In addition to the above heuristics, some exact solution procedures have been developed guaranteeing optimality. The major drawback of these procedures has been their inability to handle large problems without severe increases in computation time. Perhaps the first appearance of an exact procedure was by Balinski and Quandt [2], who utilized a version of Gomory's [22] cutting plane method for integer solutions. They were, however, severely limited by computational time and the largest problem tested had only 15 demand points. Pierce [37] developed an exact procedure which seems an improvement, but is also severely limited by size. The largest problem he reports results for has 19 demand points; a problem in which computation time was so prohibitive, the procedure was terminated at 40 minutes without guaranteeing optimality. Dun [14] developed a two-phase algorithm in which the first phase generates and prices feasible routes, and the second phase selects an optimal set of these routes. While Dun's computational results are improvements in some cases (compared to other exact procedures), often solution time is longer than previous methods.

Perhaps the most promising exact technique to date is that developed by Svestka and Huckfeldt [40]. Their procedure is designed to solve a closely related m-salesman traveling salesman problem. The procedure first solves a related assignment problem using a modified transportation algorithm of Ford and Fulkerson's [18]. If the solution to the modified assignment problem does not satisfy the conditions of the m-salesman problem, they impose restrictions upon the cost matrix, D. These restrictions prohibit the formation of the previous solution, but do so without prohibiting any feasible solutions. In this method the assignment problem is resolved until all conditions are satisfied. Forming a branch of a solution tree with the sequence of assignment problems, a branch and bound technique is used to guarantee optimality. They report solutions to a one-salesman problem of size 60 in approximately 80 seconds (using a Univac 1108 machine). Orloff [36] has suggested (although no computational results are available) a procedure similar to this. Bellmore and Malone [7] have discussed many of the principles used by both Orloff and Svestka and Huckfeldt. However, Svestka and Huckfeldt have presented the most encouraging results. While, as they state, their solution time does increase exponentially with size (implying that at some point solution times will become prohibitive) this rapid increase is delayed long enough to permit solution of relatively large problems (60-80 demand points) before becoming restrictive. Figure 4 shows how increasing problem size affects their solution time.

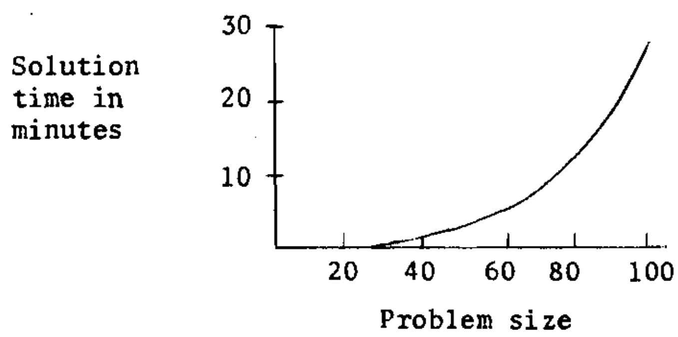


Figure 4. Solution Times of Svestka and Huckfeldt

They estimate solution time for a problem of size 100 to be approximately 27 minutes. Other interesting results are: their extension to non-symmetric cost matrices; their conclusion that the one-salesman traveling salesman problem is more difficult than the m-salesman traveling salesman problem; and that there appears to be an optimal range, $3 \leq [n/m'] \leq 7$ (the average number of demand points per route), for the number of salesmen to reduce computational time.

Unfortunately, a few considerations render this procedure unfavorable for the problem as defined within the framework of the United States Postal Service. First, the problem size necessary (100 demand points) still confronts some time difficulties (see Figure 4). Secondly, the number of salesmen (or vehicles) must be known before solving the problem. This does not allow one to explicitly consider the truck capacities. This specifies that all demands must be serviced by a predetermined number of trucks. Third, due to capacity restrictions, the minimum number of vehicles required, as determined by total capacity, may not be able to service all demands. (For example five demands of two units each cannot be serviced by two trucks of capacity five each. Each truck could service only two demands, four units, for a total of eight units, leaving two units unserviced.) Further, the minimum number of trucks capable of satisfying all demands may not minimize the total cost. Hence, the selection of the number of vehicles to be used before solving the problem may not minimize cost. Solutions for more than one m' must be determined to assure optimality (see Orloff [36]). In addition, the ease of solution is to some degree dependent upon the number of salesmen in a solution.

Unfortunately, then, despite the encouragement shown by the above results, the problem size and the relatively frequent solutions required by the United States Postal Service still seem to limit solution techniques to heuristic procedures.

Overview of Existent Heuristic Techniques for Application to Non-Symmetric Problems

In considering a cost matrix, D, if distance is the system measure used, symmetry is often a valid assumption. Since in most instances the same routes would (or could) be followed, differences are slight. However, in most measures considering actual cost, significant differences can appear. An example is the partial cost matrix shown in Table 1 for travel times between various points in the Atlanta metropolitan area. Variations greater than 50 percent are shown. This situation is typical of Postal Service collection times. Differences in time would dictate similar differences in cost because of gas, vehicle utilization, driver utilization, etc. These differences would be of concern, whenever they occur, in order to obtain a realistic cost minimization procedure. The nonsymmetry summarized in Table 1 is primarily caused by traveling with or against main traffic streams (but along the same routes). This phenomenon may be anticipated in any intra-city travel.

Table 1. A Sample Non-Symmetric Cost Matrix

	1	2	3	4	5	6
1	--	20	20	20	35	20
2	20	--	15	20		
3	25	20	--		25	
4	30			--		10
5	20		15		--	
6	25			5		--

However, there are many additional possible sources of non-symmetry, i.e., road obstacles, road repairs, travel by boat downstream and car upstream, travel up-hill in one direction and down-hill in the other direction, travel fully loaded and unloaded, all of which could create significant nonsymmetry. The symmetric problem is a special case of the non-symmetric problem. Any special circumstances could cause enough deviation from symmetry to require non-symmetric analysis. Most realistic assessments of cost would include nonsymmetry, yet all procedures (with the exception of Svestka and Huckfeldt) mentioned in the previous section, do not consider this case. In view of the numerous situations mentioned above which could incur non-symmetric costs, the need for an efficient solution procedure is evident. Existing procedures will be examined towards this end.

A cursory review of the existing heuristic techniques mentioned above does not seem encouraging for solution of problems with non-symmetric cost matrices.

Some procedures (Hayes, Christofides, and Gillette) are dependent upon Euclidean distances for solution formulation. This dictates symmetric cost.

The procedures presented by Dantzig and Ramser and Clarke and Wright, seem also dependent upon a symmetric cost matrix. As Hayes [23] states, "Although Dantzig and Ramser do not specifically mention the problem of nonsymmetry, it appears that their method would need considerable alteration in order to handle it." Tillman and Cochran [42] echo Hayes with regard to Clarke and Wright's procedure and their own. They state, "Neither method guarantees an optimal solution and both require the distance matrix to be symmetric." Other researchers, using Clarke and Wright's or other procedures, always assume, implicitly or explicitly, a symmetric cost matrix. Hayes further states that the inability to handle non-symmetric problems stems from the lack of concern of directionality in solution procedure.

Clarke and Wright's procedure does have some characteristics that favor applying it to non-symmetric problems. First, their procedure seems to be the only existing heuristic procedure that is not completely dependent upon symmetric costs. They utilize symmetry in their procedure, but their savings concept can be extended to the non-symmetric case. Second, their procedure is among the best in terms of achieving fast, good quality solutions. Some procedures achieve marginally better solutions, but at the cost of computational time.

Because of the above considerations, the procedure of Clarke and Wright, shall be used as a basis for the solution of vehicle scheduling problems having non-symmetric cost matrices. In Chapter II we shall show that not only are directionality considerations manageable, but in so doing the solution procedure is simplified with corresponding reductions in computation time.

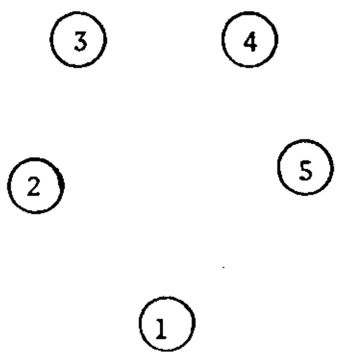
CHAPTER II

DEVELOPMENT OF A HEURISTIC SOLUTION PROCEDURE

Review of Clarke and Wright Symmetric Procedure

Because the method of Clarke and Wright [10] serves as a basis for a non-symmetric algorithm, a review of their procedure is warranted. This shall be done readily by a sample problem.

Assume the following information is given with reference to the problem depicted in Figure 5.



(a)

Point	Demand
1	0
2	5
3	15
4	10
5	5

(b)

D =

	1	2	3	4	5
1	--	10	20	20	10
2	10	--	10	20	10
3	20	10	--	15	20
4	20	20	15	--	10
5	10	10	20	10	--

(c)

=

1				
10	2			
20	10	3		
20	20	15	4	
10	10	20	10	5

(d)

Figure 5. The Cost Matrix Used by Clarke and Wright

The demands (Figure 5b) at points (2), (3), (4), and (5) are to be collected using the available vehicles located at the terminal (1). Each point can be visited once and only once. Available are one truck of capacity 20 units and a number of trucks with a capacity of 15 units. The cost matrix, D, associated with travel between these points is given. It is assumed that the cost matrix, D, is symmetric (Figure 5c) and therefore it can be represented by the half matrix shown in Figure 5d. For example, the cost of travel from point 3 to point 2 or from point 2 to point 3 is found in cell (3,2), $d_{3,2} = 10$. The savings for each possible joining of demand points is calculated from $s_{i,j} = s_{j,i} = d_{1,i} + d_{1,j} - d_{i,j}$ and a savings matrix formed (Figure 6a). Since the savings matrix is symmetric, it can also be represented by a half matrix as shown.

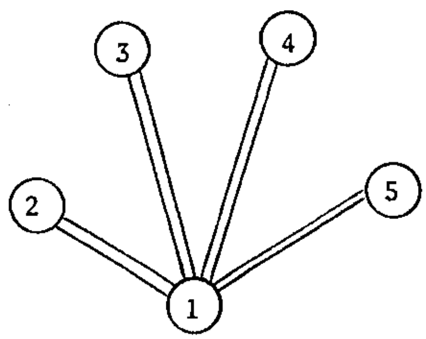
S =

2			
20	3		
10	25	4	
10	10	20	5

(a)

1				
-2	2			
-2	20	3		
-2	10	25	4	
-2	10	10	20	5

(b)



(c)

Figure 6. The Savings Matrix and Initial Solution Used by Clarke and Wright

An initial feasible solution is formed by assigning a truck, of smallest feasible capacity, to each point. This is indicated in the initial solution matrix by setting all $x_{i,1} = -2$ (Figure 6b). Each cell in the solution matrix is used to indicate whether or not the two points represented by that cell are joined at this state of the solution. Initially setting

$$x_{i,1} = -2 \qquad \text{for } i = 2,\ldots,n$$

indicates that both the path from i to 1 and from 1 to i are assigned (Figure 6c). Setting $x_{i,j} = -1$ means that either $x_{i,j}$ or $x_{j,i}$ is assigned. It is further required that at each state of the solution; i.e., after each joining, it must hold that

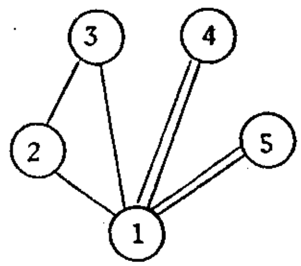
$$x_{i,k} + x_{k,j} = -2 \quad \text{for } k = 2,\ldots,n \ \forall_{i,j}$$

This is analogous to requiring the incidence of each point be equal to 2, or subsequently that each demand point be assigned to one and only one truck.

Joinings are then made by referring back to the savings matrix. The pair having the largest savings and satisfying all other constraints is joined. When a joining is made, the cell of the solution matrix which represents the two points joined, is marked with a -1. At the same time +1 is added to the first column of each row that represents a point in the joined pair. When the first column of a row = 0 in the solution matrix, it means that a demand point is joined to two other demand points. At this point that row is no longer examined in the savings matrix for possible joining, because a pair once joined, remains joined.

	1	2	3	4
2	-1			
3	-1	-1		
4	-2	10	0	
5	-2	10	10	20

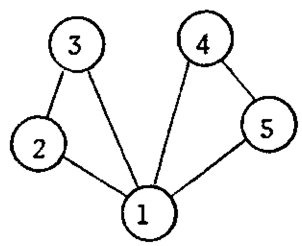
(a)



(b)

	1	2	3	4
2	-1			
3	-1	-1		
4	-1	0	0	
5	-1	0	0	-1

(c)



(d)

Truck Size	Number Available	Assigned Step 1	Assigned Step 2
20	1	1	1
15	>4	0	1

(e)

Figure 7. Intermediate and Final Solutions of Clarke and Wright

In the example of Figure 7, the pair of points having the maximum possible savings, (3,4), is then tested for joining. Here the total capacity of points 3 and 4 is 25 units. This exceeds the largest truck capacity. This joining is therefore infeasible and these points are never again considered for possible joining. The pair of points having the next largest savings (2,3) is examined. The total demand at these points, 20, exceeds that of the smaller trucks, but can be met by the larger truck. This joining is the assigned as shown in Figures 7a and 7b. One is added to the first column of the row of each point in the joined pair and a -1 marked in the (2,3) box to indicate that points 2 and 3 have been joined. The current solution is shown in Figure 7a, with a "0" in cell (3,4) to indicate that the joining of these points is infeasible. The next largest savings, cell (4,5) is tested, found to be feasible, and therefore joined. All other cells are found to be infeasible. The final solution is represented in Figures 7c and 7d. The final solution consists of two routes, one from 1-2-3-1 and the other from 1-4-5-1 at a total cost of (10 + 10 + 20) + (20 + 10 + 10) = 80 units.

The formulation of Clarke and Wright's procedure can then be summarized as follows:

(1) Minimize $$-\sum_{i=1}^{n}\sum_{j=1}^{n} d_{i,j}\, x_{i,j} \qquad (1)^*$$

Subject to

(2) $$\sum_{i=1}^{n} x_{i,k} + \sum_{j=1}^{n} x_{k,j} = -2 \qquad k = 2,3,\ldots,n \qquad (n-1)$$

(3) $$\sum_{i=z}^{n} x_{i,1} = -2m' \qquad (1)$$

---

*Note: The $x_{i,j}$ have been assigned nonpositive numbers to clarify their meaning in the savings matrix. Here, as later, a term involving $x_{i,j}$ must be negated to give the term a positive value.

$$(4)\quad x_{i,j} = 0 \text{ or } -1 \qquad j = 2,3,\ldots,n \quad i = 1,2,\ldots,n \qquad (n^2-2n+1)$$

$$x_{i,1} = 0,\ -1,\text{ or } -2 \qquad i = 2,3,\ldots,n \qquad (n-1)$$

$$(5)\quad \sum_{\ell=1}^{tz} q_i \leq C_z \qquad z = 1,2,\ldots,m' \qquad (m)$$

The term in parenthesis at the right indicates the number of each type of constraint.

In the above formulation (1) determines the total cost. The constraints (2) specify that each demand point must be serviced and that each is serviced only once. The constraint (3) specifies that each of the m routes must be made to and from the terminal. The constraints (4) limit each truck to one trip and each demand point to one visitation. The effect of (3) and (4) concurrently is to eliminate subtours (here defined as a closed route of more than one point not including the terminal). The constraints (5) prohibit vehicle capacities from being exceeded.

At the conclusion of Chapter I, rationale was presented for selecting Clarke and Wright's procedure as a basis for a non-symmetric vehicle scheduling algorithm. In addition to those reasons, others can be derived from the use of the "savings" concept.

The idea behind savings is simple. Joining two points on a route will reduce the total cost required to service those points. The cost reduction will be equal to the savings associated with those two points. Thus, the total cost after joining two points is equal to the total cost before joining, less the savings of the joined pair. That is,

$$\text{Cost (after)} = \text{Cost (before)} - \text{Savings}$$

This relation is true for each pair of points joined. Then, it must also be true that the final solution must equal the initial solution less the sum of all savings obtained by joining points.

$$\text{Cost (final)} = \text{Cost (initial)} - \left(- \sum_{i=2}^{n} \sum_{j=2}^{n} s_{i,j} \; x_{i,j} \right)$$

Utilizing this relationship, the total cost at any stage of the solution process can be determined. Also, the final cost can be determined with only a knowledge of the initial cost and the savings of joined points. For the example of Figure 6, the initial cost is 120 units. Two pairs of points are joined, each with a savings of 20. The final cost is then 120 - 20 - 20 = 80. Determining the cost in this manner is easier than using inter-point distances. It is also more useful in that the cost at intermediate steps is always known.

The same relation between initial and final costs holds for any solution, not only those obtained by the Clarke and Wright procedure. Thus, for the optimal solution, it must be true that

$$\text{Cost (optimal)} = \text{Cost (initial)} - \left(- \sum_{i=2}^{n} \sum_{j=2}^{n} s_{i,j} \; x_{i,j} \right)$$

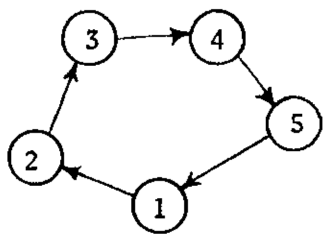
i.e., the optimal solution maximizes total savings. This is encouraging since Clarke and Wright's procedure is an attempt to maximize savings. This shows that the optimal solution can always be found from the savings matrix.

Modifications to the Basic Procedure

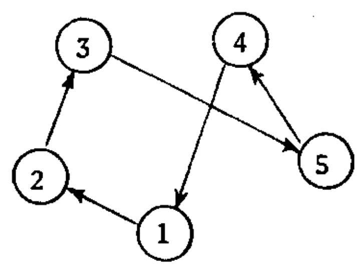
The savings concept is useful since it obtains the current solution cost in a simplified manner. It has also been shown to be capable of finding the optimal solution. For these reasons, and those mentioned in Chapter I, the Clarke and Wright method is favorable for extension to non-symmetric problems. However, some modifications are needed because of the increased complexity of the cost matrix and because directionality must be considered. The modifications necessary and corresponding procedural changes are best illustrated by an example.

An example proposed by Hayes [23] shall be used to show the inadaptability of the Dantzig and Ramser procedure to non-symmetric cost matrices. Assume the partial cost matrix as shown in Figure 8.

	1	2	3	4	5 ...
1	--	20	20	20	20
2	20	--	3	30	35
3	20	25	--	20	30
4	20	30	20	--	18
5	20	35	30	4	--
⋮					



Dantzig and Ramser Solution

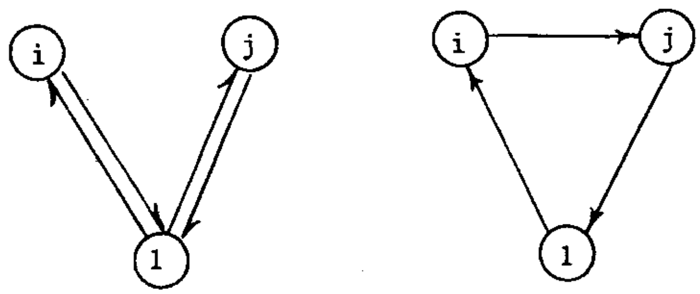


Optimal Solution

Figure 8. A Non-Symmetric Problem Where the Dantzig and Ramser Procedure Fails to Find the Optimal Solution

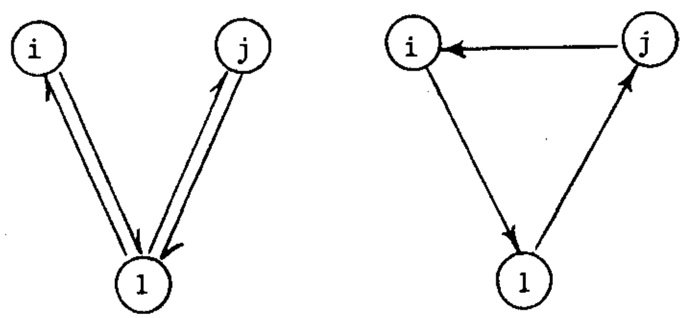
The distance matrix is twice as large for non-symmetric problems since $d_{i,j} \neq d_{j,i}$ and each has a distinct value in the matrix. Strict use of the Dantzig-Ramser procedure obtains a route assignment of 1-2-3-4-5-1 with a cost of 81. Hayes points out a better solution would be a route of 1-2-3-5-4-1 with a cost of 77.

To solve the problem with a non-symmetric distance matrix, full matrices are used and certain aspects of the procedure are modified. Using the given cost matrix, a savings matrix (Table 2) is again calculated. This time directionality in savings calculations is accounted for by:



$$s_{i,j} = d_{i,1} + d_{1,j} - d_{i,j}$$

and now



$$s_{j,i} = d_{j,1} + d_{1,i} - d_{j,i}$$

Figure 9. A Non-Symmetric Savings Calculation

Table 2. Non-Symmetric Cost and Savings Matrices

(a) Savings

	1	2	3	4	5
1					
2			37	10	5
3		15		20	10
4		10	20		30
5		5	10	36	

(b) Solution

--	-1	0	-1	-1
0	--	-1	10	5
-1	15	--	20	10
-1	10	20	--	30
-1	5	10	36	--

In the symmetric case $d_{j,1} = d_{1,j}$, $d_{i,1} = d_{1,i}$, etc. and $s_{i,j} = s_{j,i}$. This, as shown in Table 2a, is not true for the case of non-symmetry.

An initial solution of one truck assigned to each demand point is still assumed. This is now represented by an initial solution matrix (Table 2b) with a -1 in each cell of the first column and in each cell of the first row. This indicates that a truck is assigned from the central facility to each demand point, and that a truck is assigned from each demand point back to the terminal.

The first step is to select the pair of points having the greatest savings as a candidate for joining. These points are then tested against the various constraints for feasibility, and joined if found to be feasible. For the given example, the pair (2,3) with a savings of 37 is greatest. It is assumed that the capacity and other requirements are met. The first step solution matrix is shown in Figure 12b. The joining of pair (2,3) is denoted by placing a -1 in cell (2,3) of the solution matrix. This indicates that a truck is assigned to travel from point 2 to point 3. A 1 is then added to elements (2,1) and elements (1,3) of the solution matrix. Note that now element (2,1) = (1,3) = 0. This means that row 2 and column 3 need not be considered further for joinings. This adheres to logic. The elements of row 2 represent all possible trips leaving demand point 2. Since each demand point is to be serviced by only one truck, only one truck can leave any demand point. Therefore, once a trip from i is assigned, no more trips from i are possible. Also note that assigning a trip from i to j precludes the possibility of travel from j to i, since this would violate the constraint that each point is visited only once. Therefore, whenever arc (i,j) is assigned, $s_{j,i}$ will be set equal to zero to avoid selecting this link for joining. This restriction avoids the formation of subtours of length two.

The size of the savings matrix to be considered can thus be reduced as shown in Table 3a, eliminating row 2 and column 3 and

Table 3. Solution Matrices for a Non-Symmetric Problem

	1	2	3	4	5
1	--	1	0	1	1
2	0	--	-1	--	--
3	1	0	--	20	10
4	1	10	--	--	30
5	1	5	--	36	--

(a) Reduced Savings Matrix

	1	2	3	4	5
1	--	-1	0	0	-1
2	0	--	-1	--	--
3	-1	0	--	--	10
4	-1	10	--	--	0
5	0	--	--	-1	--

(b) Intermediate Solution Matrix

	1	2	3	4	5
1	--	-1	0	0	0
2	0	--	-1	--	--
3	0	0	--	--	-1
4	-1	10	--	--	--
5	0	--	--	-1	--

(c) Final Solution Matrix

setting $s_{3,2} = 0$. This reduction in the size of the savings matrix is highly favorable from a computational standpoint. The number of entries of the savings matrix which must be searched decreases rapidly. This indicates that despite the increased size of the distance matrix, the solution might actually be less complicated and involve less computation time. This is verified experimentally in Chapter III. This finding is comparable to results of Bellmore and Malone [7] and Svestka and Huckfeldt [40], although they each use exact solution procedures. Their results show that problems with non-symmetric cost matrices are less complicated and involve less solution time than problems with symmetric cost matrices.

Continue using the reduced savings matrix. The greatest feasible savings at this step is 36 for the pair (5,4). Join this pair and continue in a manner similar to that described for the first step. Joinings are made until no more are possible. Assuming no capacity restrictions are encountered in this problem, the final solution matrix of Table 3c is obtained. A route of 1-2-3-5-4-1 with a cost of 20 + 3 + 30 + 4 + 20 = 77 arises. This agrees with Hayes' solution.

The creation of subtours of length greater than two should be avoided. This can be done explicitly, each time a joining is made, by setting the savings of pairs forming subtours equal to zero. In Figure 10 if point 5 is joined to the existing route, 1-2-3-4-1, one would set not only $s_{5,4} = 0$, but also $s_{5,3} = s_{5,2} = 0$.

It is also possible to avoid this problem by ascertaining that the two points selected for joining have not already been assigned to the same route. In this procedure, the latter method has been utilized

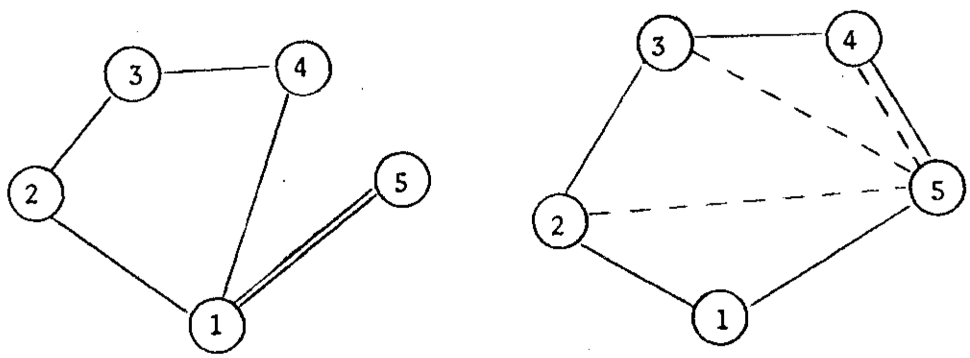


Figure 10. Eliminating Subtours

because it is compatible with the computer logic necessary for other calculations. This method does not preclude combining existing routes. Two points such as 3 and 4, in Figure 11, may still be joined even though both have already been assigned to routes.

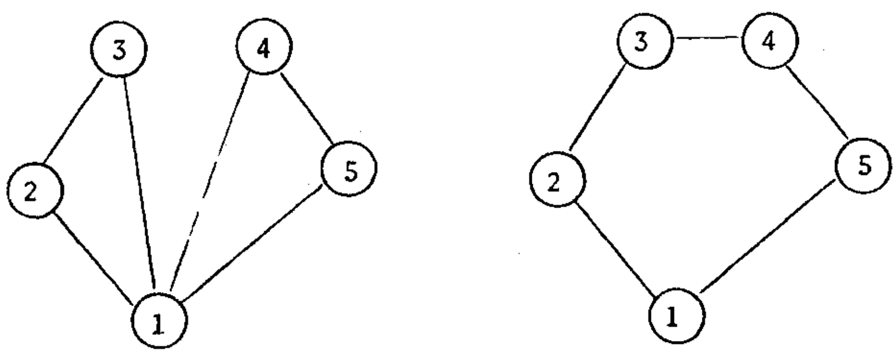


Figure 11. The Joining of Two Routes

A revised formulation for the general solution procedure is:

(1) Minimize $$- \sum_{i=1}^{n} \sum_{j=1}^{n} d_{i,j} \; x_{i,j} \qquad (1)$$

subject to

$$(2)\quad \sum_{i=1}^{n} x_{i,k} = -1 \qquad k = 2,3,\ldots,n \qquad (n-1)$$

$$\sum_{j=1}^{n} x_{k,j} = -1 \qquad k = 2,3,\ldots,n \qquad (n-1)$$

$$(3)\quad \sum_{i=2}^{n} x_{i,1} = -m' \qquad (1)$$

$$\sum_{j=2}^{n} x_{1,j} = -m' \qquad (1)$$

$$(4)\quad x_{i,j} = 0 \text{ or } -1 \qquad \forall_{i,j} \qquad i \neq j \qquad (n^2-n)$$

$$(5)\quad \sum_{\ell=1}^{tz} q_{\ell} \leq C_z \qquad z = 1,2,\ldots,m' \qquad (m')$$

In the above formulation, (1) determines the total cost of the solution. The constraints (2) assure that only one truck leaves each demand point and only one truck enters each demand point. The constrains (3) dictate that m' trucks must leave the terminal and that m' trucks must enter the terminal. The constraints (4) permit only one traversal of a directed arc between any two points. The constraints (5), as defined above, assure that the sum of the capacities of all demand points assigned to the zth route do not exceed the capacity of the truck assigned to that route.

The procedure outlined above obtains good solutions to vehicle scheduling problems that have a non-symmetric cost matrix. These solutions are, in general, suboptimal. The above procedure suffers from the same faults as does the Clarke and Wright procedure for symmetric problems: (1) failure to reconsider points once joined, and (2) demands are not matched directly with capacities.

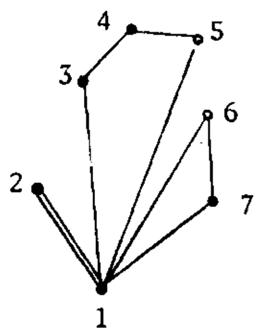
The consequences of these faults can be significant. Tillman and Cochran [42] and Hayes [23] present three example problems where use of the traditional Clarke and Wright procedure does not produce the optimal result. Each of these problems is solved in Chapter III. Inspection of why the procedure failed should provide insight into how a better solution can be obtained. For this purpose each of the three problems, the Clarke and Wright solution, and the optimal solution are reproduced here in Figures 12 through 14.

In each example problem, a pair of points selected for joining by the Clarke and Wright method does not appear at all in the optimal solution. This can be attributed, at least in the given examples, to improper matching of demands to capacities. Selection of pairs for joining entirely on the basis of savings may cause vehicles to be loaded less near capacity, or it may prohibit joinings at later stages which would result in greater total savings. For example, in Hayes' first example, Figure 12, selection of the pair (4,5) for joining evokes a solution in which 3 vehicles are utilized, only one at capacity. But if link (4,5) is suppressed ($s_{4,5} = 0$) and not considered for joining, the optimal solution is realized. This solution utilizes only 2 vehicles at their full capacity.

Tillman and Cochran's look-ahead procedure would also produce

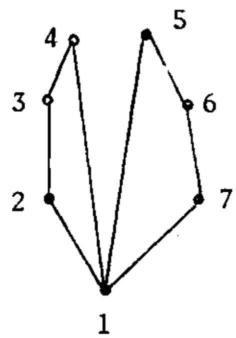
Cost Matrix

	1	2	3	4	5	6
2	10					
3	20	12				
4	25	20	10			
5	25	25	11	2		
6	20	30	22	11	10	
7	10	20	30	35	20	12



Clarke & Wright Solution
Cost = 119

	1	2	3	4	5	6
2	-2					
3	-1					
4	0		-1			
5	-1			-1		
6	-1					
7	-1					-1



Optimal Solution
Cost = 114

	1	2	3	4	5	6
2	-1					
3	0	-1				
4	-1		-1			
5	-1					
6	0				-1	
7	-1					-1

Figure 12. Hayes' First Example of a Suboptimal Clarke and Wright Solution

a suboptimal solution to this problem. Their procedure would determine that the successive joinings of demand pairs (4,5) and (5,6) maximizes total savings. The demand points (4,5) would be joined, and the optimal solution would not be achieved. The look-ahead technique stresses the order in which pairs are joined. However, it seems more likely that suboptimal solutions contain links that do not best utilize truck capacities; i.e., some links lead to suboptimal solutions because of truck capacity considerations.

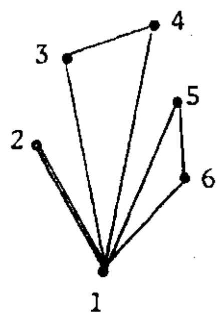
This idea is the basis for a suggested improvement to the Clarke and Wright procedure. If some links selected by the Clarke and Wright procedure are not favorable, "suppression" of these links will lead to improved solutions. The suggested improvement then, is to "suppress" certain links within the savings matrix. Suppression of a link means forcing some constraint upon the solution procedure prohibiting that link from appearing in the solution. Suppression of the link (i,j) is accomplished by setting $s_{i,j} = 0$ in the original savings matrix. This means that the pair of points, i and j, will never be selected for joining and can never enter the solution.

The optimal solution to all three example problems can be obtained from the suppression of the savings of a possible joining. In Hayes' first example, discussed above, this pair is (3,4). For Hayes' second example problem, shown in Figure 13, the link which needs to be suppressed is (3,4). For the example problem given by Tillman and Cochran (Figure 14), suppression of the pair (4.6) leads to the optimal solution.

Selection of the links to be suppressed, a priori, seems by no

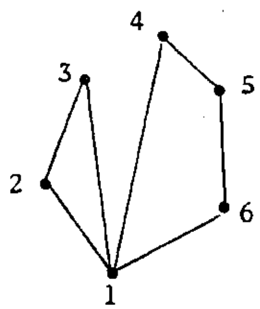
Cost Matrix

	1	2	3	4	5
2	12				
3	15	10			
4	15	11	2		
5	14	17	9	7	
6	10	18	13	12	7



Clarke & Wright Solution Cost = 86

	1	2	3	4	5
2	-2				
3	-1				
4	-1		-1		
5	-1				
6	-1				-1



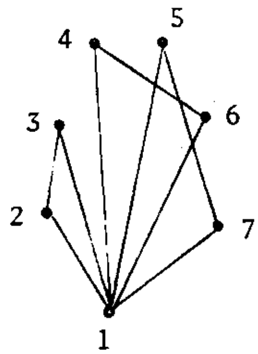
Optimal Solution Cost = 75

	1	2	3	4	5
2	-1				
3	-1	-1			
4	-1				
5	0			-1	
6	-1				-1

Figure 13. Hayes' Second Example of a Suboptimal Clarke and Wright Solution

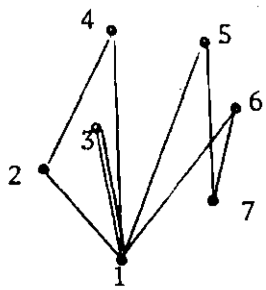
Cost Matrix

	1	2	3	4	5	6
2	38					
3	42	35				
4	56	22	33			
5	63	54	22	45		
6	64	38	27	20	29	
7	80	62	38	65	22	45



Tillman and Cochran Solution
Cost = 420

	1	2	3	4	5	6
2	-1					
3	-1	-1				
4	-1					
5	-1					
6	-1			-1		
7	-1				-1	



Optimal Solution
Cost = 394

	1	2	3	4	5	6
2	-1					
3	-2					
4	-1	-1				
5	-1					
6	-1					
7	0				-1	-1

Figure 14. Tillman and Cochran Example of a Suboptimal Clarke and Wright Solution

means a simple matter. For example, one would assume that if the demand pairs possessing the highest possible savings were joined, filling a truck to capacity, this route would appear in the optimal solution. This situation is shown in Table 4, with the associated cost and savings matrices.

Table 4. Cost and Savings Matrices Which Lead to a Suboptimal Clarke and Wright Solution

Demand	Point
4	2
4	3
5	4
3	5
4	6
3	7
3	8
5	9

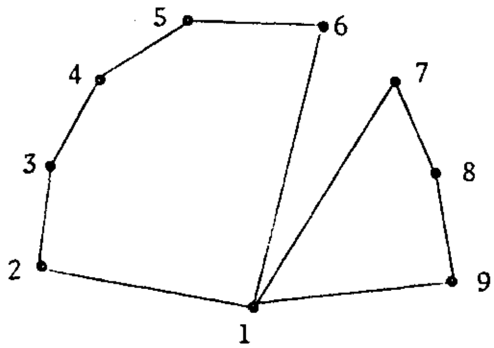
	1	2	3	4	5	6	7	8
2	10							
3	30	20						
4	40	40	50					
5	40	45	65	70				
6	50	55	75	85	80			
7	40	45	65	75	75	89		
8	30	35	55	62	62	79	62	
9	10	15	35	45	44	59	49	38

Cost Matrix

	1	2	3	4	5	6	7	8
2								
3		20						
4		10	20					
5		5	5	10				
6		5	5	5	10			
7		5	5	5	5	1		
8		5	5	8	8	1	8	
9		5	5	5	6	1	1	2

Savings Matrix

If the demands are to be serviced with one truck of capacity 20 and one truck of capacity 12, the Clarke and Wright solution is as shown in Figure 15. It contains one route, 1-2-3-4-5-6-1, filling the larger truck exactly to capacity and using the maximum possible savings.

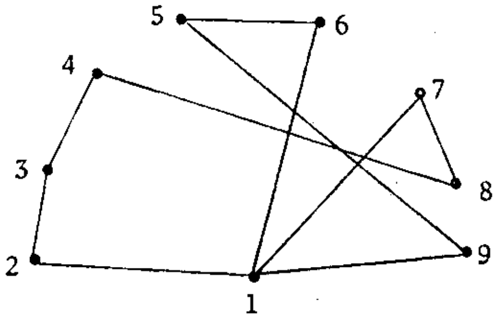


	1	2	3	4	5	6	7	8
2	-1							
3	0	-1						
4	0		-1					
5	0			-1				
6	-1				-1			
7	-1							
8	0						-1	
9	-1							-1

Figure 15. Clarke and Wright Solution to a Sample Problem

The total cost of this solution is 430. Yet, if the link for pair (4,5) is suppressed, the optimal cost of 428 is obtained with the truck of capacity 20 filled below capacity (Figure 16). Hence, neither filling a truck to capacity nor utilizing the maximum savings will guarantee optimality.

It is also true that there is no guarantee that if suppression of one demand pair results in improvement, suppression of a different demand pair would not result in greater improvement. In addition, suppression of two links between separate demand pairs simultaneously may lead to improvement when suppression of either link alone would not. This can be seen in another example problem of Hayes' shown in Figure 17 which Hayes used in a different context. The Clarke and Wright



	1	2	3	4	5	6	7	8
2	-1							
3	0	-1						
4	0		-1					
5	0							
6	-1				-1			
7	-1							
8	0			-1			-1	
9	-1				-1			

Figure 16. Optimal Solution to the Sample Problem

Point Demand

2	6
3	6
4	4
5	4

	1	2	3	4
2	100			
3	100	50		
4	100	60	10	
5	100	130	60	50

Cost Matrix

	1	2	3	4
2	-2			
3	-2	150		
4	-2	140	190	
5	-2	70	140	150

Savings Matrix

	1	2	3	4
2	-1			
3	-1	150		
4	-1	140	-1	
5	-1	-1	140	150

Clarke and Wright Solution
Cost = 540

Figure 17. Hayes' Third Example of a Suboptimal Clarke and Wright Solution

solution of 540 cannot be bettered by suppression of only one link of the savings matrix. Here, the simultaneous suppression of links (3,4) and (4,5) will give the optimal solution of 520 as shown in Figure 18.

	1	2	3	4
2	-2			
3	-2	150		
4	-1	140	--	
5	-1	70	140	-1

Solution with (3,4)
Suppressed Cost = 650

	1	2	3	4
2	-1			
3	-1	150		
4	-1	-1	--	
5	-1	70	-1	--

Optimal Solution with (3,4) and (4,5)
Suppressed Cost = 520

Figure 18. Solutions for Hayes' Third Example Using Suppression

Despite the vagaries involved in determining where this proposed suppression technique will be effective, some statements can be made to enhance its effectiveness. Suppression should only be considered for point pairs which were joined in the original Clarke and Wright-based procedure. Since this solution technique is still based on their selection process, suppressing other links would not affect the solution. Also, the links to be suppressed should be chosen from among the first selected to be joined by the solution procedure. These joinings will involve greater savings, and therefore suppression of these joinings will have a greater effect upon the solution. When an improved solution is obtained from the suppression of a link, the improved solution is saved and the previous solution discarded. The suppressed link which led to the improvement remains suppressed in all future attempts at improvement. This was done rather than suppressing all joined pairs of a solution and saving the suppression giving the maximum improvement. It was felt the latter method resembled total enumeration. The latter method would also not provide the variation of solutions produced by the method used. Therefore, when an improved solution is obtained, this solution, with the suppressed link, is saved as the new best. Further suppressions are then attempted on this new solution.

The above considerations are summarized for efficient use of this suppression technique. The problem is first solved using the basic solution procedure (Clarke and Wright for the symmetric problem or the modified procedure presented here for the non-symmetric problem), keeping a list of the order in which each joining was made. The first joined pair is then suppressed and the modified problem solved again by the original procedure. If an improvement is obtained, the new solution is saved and the procedure continues, suppressing the first joining of the new solution. If no improvement is obtained, the current suppression is voided and the next joined pair is suppressed. The process continues until a predetermined number of successive suppressions brings no improvement and is then terminated. In the larger problems, five suppressions has been found to be effective.

The suppression technique can be viewed as examining complete branches of a solution tree. In Figure 19, the ordered pairs inside the circles represent the selection of that demand pair for joinings. The heavy bar over an ordered pair represents all solutions not containing the joining of that pair.

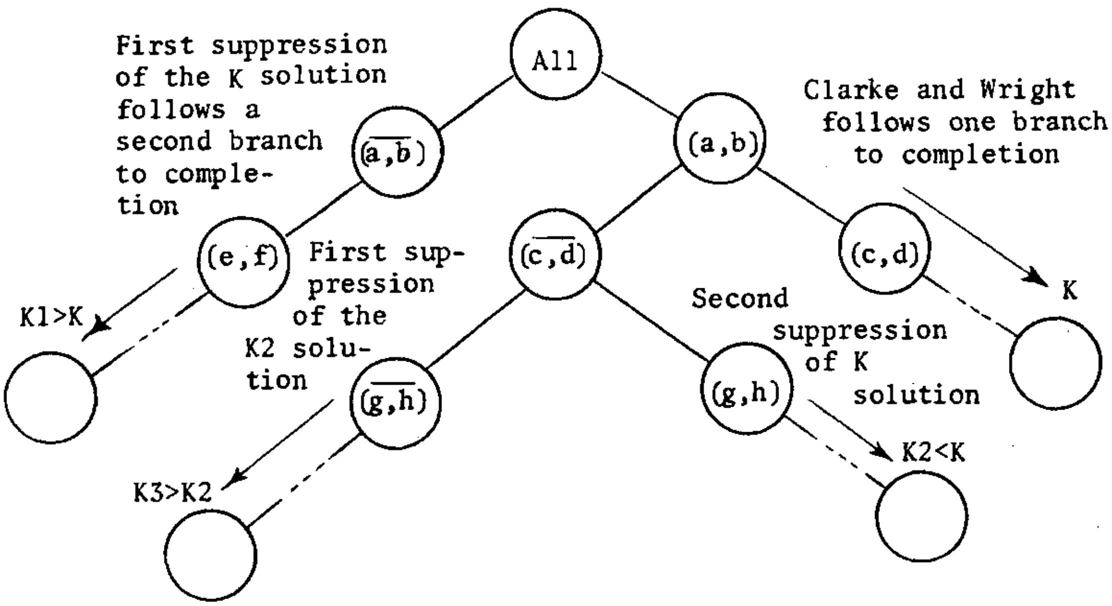


Figure 19. Solution Tree Representation of the Heuristic Procedure

The Clarke and Wright solution explores one "branch" of the solution tree to termination. This is shown as the rightmost "branch" of the tree, terminating with a solution cost of K. Each point pair within a circle ((a,b), (c,d), etc.) indicates that pair has been joined on a route in the solution represented by that branch.

After obtaining a complete solution, one backtracks to the point where this branch started. The first point pair joined, (a,b), is then suppressed ($s_{a,b} = 0$). This is denoted by $(\overline{a,b})$ to indicate all solutions which do not join the points a and b, in that order, on a route. The best solution is then found from this modified initial savings matrix. This first suppression of the K solution proceeds down the leftmost branch, terminating with a cost of K1. In the illustration, K1 is shown greater than K so the K solution is saved.

Since no improvement was obtained in the first suppression, the second pair joined in the K solution, (c,d), is suppressed. This suppression is represented by the cell $(\overline{c,d})$. All solutions from this point would contain (a,b), but not (c,d). The best solution is again found using the modified initial savings matrix. The next two points joined are (g,h) and the resulting solution is shown having a cost, K2, less than K. Since K2 is less than K, this new solution is now saved and the K solution discarded. The next step involves suppressing the first pair joined in the K2 solution, (g,h), and so the process continues.

Each time a link is suppressed a new, complete solution is found. Suppressing more links increases the chances of improving the solution. However, to guarantee an optimal solution would involve total enumeration of all solutions. A trade-off is evident between increased computational time and quality of solution. For each suppression the problem must be resolved. Despite this limitation, the results, using few suppressions, seem encouraging (see Table 7).

Summary of the Solution Procedure and An Example

Now a step-by-step computational solution procedure can be given for the non-symmetric vehicle scheduling problem which includes the suppression technique.

Step 1: Initialize the cost matrix, the demand at each point and the size and the number of trucks available.

1.1. Set the suppression level, L, equal to 1

$$L = 1$$

Step 2: Compute the savings, set up the matrix and determine the initial solution.

2.1 Compute all $s_{i,j} = d_{i,1} + d_{1,j} - d_{i,j}$ for all $i = 2,\ldots,n$, $j = 2,\ldots,n$ and $i \neq j$. If $s_{i,j} < 0$, set $s_{i,j} = 0$.

2.2 Set

$$s_{i,1} = s_{1,j} = -1 \quad \text{for } i = 2,\ldots,n \text{ and } j = 2,\ldots,n.$$

This determines the initial solution.

2.3 Compute the cost of the initial solution, K, by

$$K = \sum_{i=2}^{n} d_{i,1} + \sum_{j=2}^{n} d_{1,j}$$

Step 3: Determine the greatest feasible savings, $s_{\hat{i},\hat{j}}$, where

$$s_{\hat{i},\hat{j}} = \max_{\overline{(i,j)}} \{s_{i,j}\}$$

where $\overline{(i,j)}$ is the subset of all (i,j) such that $(i,1) \neq 0$ and $(1,j) \neq 0$.

3.1 If $s_{\hat{i},\hat{j}} = 0$, proceed to Step 5.

Step 4: Join the demand points $\hat{i}$ and $\hat{j}$ on a route.

4.1 Determine if either of the points is currently assigned to a route. There are three possibilities:

4.1.1 Neither of the points is on a route. Join i and j on a new route, z. Compute the total demand

$$Q_z = q_{\hat{i}} + q_{\hat{j}}$$

and proceed to Step 4.2.

4.1.2 One of the points is currently assigned to a route, z. Attempt to join the unassigned point to the route. Compute the total demand

$$Q_z = Q_z + q_{\hat{i}} \text{ (or } q_{\hat{j}}\text{)}$$

and proceed to Step 4.2.

4.1.3 Both points are currently assigned to routes, r and s. Attempt to join the routes. Compute the total demand

$$Q_z = Q_r + Q_s$$

and proceed to Step 4.2.

4.2 Check the capacity restrictions.

4.2.1 Choose the smallest C_k such that

$$C_k \geq Q_z$$

and proceed to Step 4.3.

4.2.2 If no such C_k exists set $s_{\hat{i},\hat{j}} = 0$. Return to Step 3.

4.3 Decrement the number of trucks available

$$T_k = T_k - 1$$

(When applicable, increment the number of trucks available for the previous truck size used on this route.)

4.4 Mark the points $\hat{i}$ and $\hat{j}$ as joined in the savings matrix,

$$s_{\hat{i},\hat{j}} = -1$$

$$s_{\hat{j},\hat{i}} = 0$$

$$s_{\hat{i},1} = s_{1,\hat{j}} = 0$$

4.5 Compute the new solution cost

$$K = K - s_{\hat{i},\hat{j}}$$

4.6 Save the order in which the points were joined and return to Step 3.

Step 5: Save the best solution.

5.1 If this was the first solution, save the cost, K'

$$K' = K$$

the routes, and the order in which points were joined. Proceed to Step 6.

5.2 If this is not the first solution

5.2.1 and $K < K'$, set $K' = K$, $L = 1$, and $s_{\hat{i},\hat{j}} = 0$ in the matrix of Step 2.1. Proceed to Step 5.3.

5.2.2 and $K > K'$, set $L = L + 1$ and proceed to Step 5.3.

5.3 Save the routes formed and the order in which points were joined for the best solution.

Step 6: Suppression of arcs.

6.1 If $L < L'$ (the maximum number of suppressions without improvement in the solution), suppress the pair of points joined next in the current best solution, say i' and j', by

$$s_{i',j'} = 0$$

in the savings matrix of Step 2.1 and proceed.

6.2 If $L = L'$, or if all joined pairs in the current solution have been suppressed, terminate.

The steps of this procedure will now be demonstrated with an example. Consider the non-symmetric cost matrix shown in Table 5a.

Table 5. Information for a Complete Expository Problem

	1	2	3	4	5	6	7
1	--	20	30	50	60	50	40
2	10	--	5	10	20	20	15
3	20	10	--	30	10	35	20
4	30	15	20	--	10	15	10
5	40	15	5	10	--	15	5
6	30	30	25	10	5	--	20
7	20	10	30	20	10	30	--

(a) Cost Matrix

	1	2	3	4	5	6	7
1	--	-1	-1	-1	-1	-1	-1
2	-1	--	35	50	50	40	35
3	-1	30	--	40	20	35	40
4	-1	35	40	--	80	65	60
5	-1	45	65	80	--	75	75
6	-1	20	35	70	85	--	50
7	-1	30	20	50	70	40	--

(b) Savings Matrix

Point	Demand
1	-
2	6
3	2
4	5
5	5
6	8
7	6

(c)

Number trucks	Size
1	16
∞	8

(d)

Steps 1 and 2 give the information summarized in Tables 5b through 5d. The cost, K, of Step 2.3 is the sum of all elements of the first row and first column of the cost matrix of Table 5a.

K = 10 +20 + 30 + 40 + 30 + 20 + 20 + 30 + 50 + 60 + 50 + 40 = 400.

The first pass through Step 3 gives $s_{\hat{i},\hat{j}} = 85$ with i = 6 and

$j = 5$. Proceeding through Step 4.1 it is determined that neither of the points 6 or 5 is currently assigned to a route. From Step 4.1.1 the total demand of these two points is 8 + 5 = 13 units. Since the truck of capacity 16 is available these points are joined. The truck size, total demand, and the points on this route (6 and 5) are saved. Steps 4.4 and 4.5 give the information summarized in Figure 20a.

(a)

	1	2	3	4	5	6	7
1	--	-1	-1	-1	0	-1	-1
2	-1	--					
3	-1		--				
4	-1			--			
5	-1				--	0	
6	0				-1	--	
7	-1						--

(b)

	1	2	3	4	5	6	7
1	--	-1	0	-1	0	-1	-1
2	-1	--					
3	-1		--		0		
4	-1			--			
5	0		-1	0	--	0	
6	0				-1	--	
7	-1						--

Order of joined pairs

(6,5)
(5,3)

Figure 20. Solution Matrices for an Expository Problem

Returning to Step 3, the next $s_{\hat{i},\hat{j}} = 80$ for the pair of points 5 and 4. In Step 4.1.2 a match is found since demand point 5 is currently assigned to a route. It is seen in Step 4.2 that the total demand, 13 + 5 = 18, exceeds the capacity of the largest available truck. $s_{5,4}$ is then set equal to zero and the procedure returns to Step 3. In this manner the next pair feasible to join is determined to be pair (5,3) with a savings of 65. All other point pairs are found to be infeasible and the final solution, K' = 250, is summarized in Figure 20b.

Since $s_{\hat{i},\hat{j}} = 0$, the procedure moves to Step 5. The cost, K, of 250 is better than the initial cost of 400 and the current solution is saved. In Step 6, since L = 1, the first pair of points joined in the current solution, 6 and 5, are suppressed. This gives an initial savings matrix as shown in Figure 21a.

	1	2	3	4	5	6	7
1	--	-1	-1	-1	-1	-1	-1
2	-1	--	35	50	50	40	35
3	-1	30	--	40	20	35	40
4	-1	35	40	--	80	65	60
5	-1	45	65	80	--	75	75
6	-1	20	35	70	0	--	50
7	-1	30	20	50	70	40	--

(a)

	1	2	3	4	5	6	7
1	--	-1	0	-1	0	-1	0
2	0	--	-1				
3	-1	0	--				
4	0			--	-1		
5	0			0	--		-1
6	-1				0	--	
7	-1				0		--

(b)

Order of joined pairs

(4,5)
(5,7)
(2,3)

Figure 21. First Use of Suppression On An Expository Problem

Following Steps 3 and 4 as before now results in a solution as shown in Figure 21b with K = 210. Again Step 5.2 finds K to be less than K' and the current solution replaces the previous one (K' = K). The arc $s_{6,5}$ is now permanently set equal to zero.

The first joined pair of the new solution, $s_{4,5}$ is now suppressed resulting in the initial savings matrix of Figure 22a.

Following Steps 3 and 4 now gives the solution shown in Figure 22b with K = 285. This is not less than the current best K' of 210. Hence, the new solution is ignored. The next two joined point pairs

	1	2	3	4	5	6	7
1	--	-1	-1	-1	-1	-1	-1
2	-1	--	35	50	50	40	35
3	-1	30	--	40	20	35	40
4	-1	35	40	--	0	65	60
5	-1	45	65	80	--	75	75
6	-1	20	35	70	0	--	50
7	-1	30	20	50	70	40	--

(a)

	1	2	3	4	5	6	7
1	--	-1	-1	0	0	-1	-1
2	-1	--					
3	0		--		-1		
4	-1			--	0		
5	0		0	-1	--		
6	-1				0	--	
7	-1						--

(b)

Order of joined pairs

(5,4)
(3,5)

Figure 22. Further Suppressions on an Expository Problem

in the current best solution, $s_{5,7}$ and $s_{2,3}$, are suppressed separately. For each new cost matrix a solution is found. Neither solution improves upon the current best. Since all elements of the joined point pairs have been explored, Step 6.2 directs the procedure to terminate. This gives a final solution as shown in Figure 23. The solution contains three routes: 1-2-3-1, 1-4-5-7-1, and 1-6-1. These routes can also be determined from the solution matrix of Figure 21b.

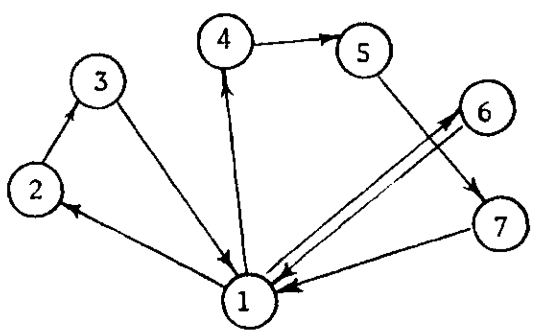


Figure 23. Optimal Solution of an Expository Problem

The steps used to arrive at the final solution are depicted in Figure 24. This figure is intended to clarify the complete procedure described verbally above. Again, as in Figure 19, the similarity to a branch and bound tree is evident.

This procedure, as presented, has several advantages. It is very easy to incorporate within existing techniques in that the same solution procedure is used. The order of joined points to be examined by suppression, is determined as a by-product of normal procedure. Also, one immediately has a good solution, Clarke and Wright's, and improvement can be extended to the desired degree.

In addition to these advantages, a number of alternate solutions are generated. This can be valuable in systems where intangible factors may not favor the best available solution. Several solutions near the best are usually obtained among the alternates produced by the suppression of links. For example, in a randomly generated non-symmetric problem with 40 demand points, 5 of the 8 solutions generated were within 10 percent of the best heuristic solution obtained. The alternate solutions produced may differ in the number of trucks assigned, the unused capacity of trucks assigned, and the sizes of trucks assigned. All of these factors could be of value.

In addition to the good quality, alternate solutions generated, a few of poor quality are generated. This identifies these links which are critical to the system. Knowing which links most affect the total system cost can be helpful, both in considering immediate changes and planning for the future.

The use of a suppression technique can thus be useful as a tool

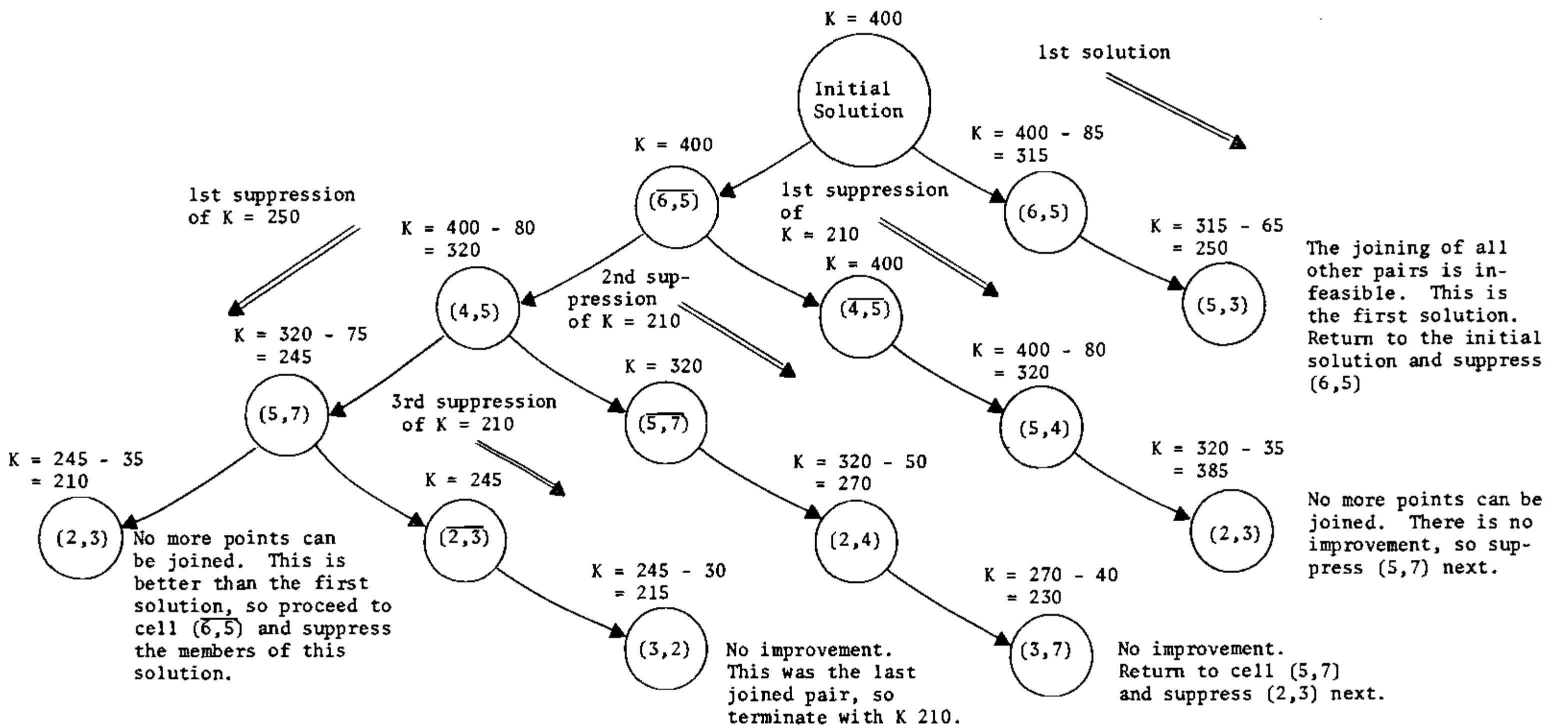


Figure 24. The Heuristic Solution Procedure Presented as a Tree

in system analysis. It has the additional advantages of providing alternate solutions, and providing better quality solutions.

Additional Constraints

Care must be taken in considering additional constraints with a solution procedure of this type. The primary reason for selecting an heuristic procedure was the ease and speed of solution generation. The addition of certain constraints can increase the problem complexity so that the advantage of using an heuristic procedure (quick solutions to large problems) is lost. Various possible system constraints shall be discussed with this idea in mind.

Limiting the total cost incurred by any given vehicle on a route could be easily incorporated. The cost of a route can be determined easily at each step as a by-product of other calculations. However, within the system of the United States Postal Service, truck capacities are filled long before vehicle cost restrictions are reached.

Another constraint could be that some demands are so large that they cannot be serviced entirely by one truck. Demands are not considered explicitly when selecting points for joining, and therefore, the demands can be considered variable. When such a point is considered, the maximum demand available on a truck will be assigned. The remaining demand at that point will still be available for joining in the savings matrix. This procedure would continue until all demand at the point is serviced. This situation is encountered by the United States Postal Service. However, the United States Postal Service sends frequent trips to areas of high demand. What is not serviced on a first trip is serviced on subsequent trips. They do not operate under the constraint that all demand at a point must be serviced at one time. This being true, it can be realistically assumed that the greatest demand is equal to one full truck.

Another consideration could be that some points both receive and ship goods. Truck capacities could thus be reached either by the goods it must deliver or the goods it must pickup. While this is true of the United States Postal Service, one activity usually dominates. In the morning almost all goods are to be delivered to points, and in the evening, almost all goods are received. The problem could then be handled as two separate problems, one delivery problem and one collection problem.

A final, very real constraint of the United States Postal Service is that many demands must be serviced within specified time periods. In addition, many of these demands must be serviced more than once a day. This constraint was not included for two reasons. First, throughout most of the day, the total system demand is low and these considerations do not become a problem. Second, these considerations greatly complicate the solution process. Work on this type of problem has been done by Levin [27] and Martin-Loef [31], among others. However, inclusion of this type of analysis with that presented above would lose any advantage gained by simple heuristic algorithms.

From the preceding discussion of additional constraints, it is evident that the formulation given above of the non-symmetric vehicle scheduling problem provides a realistic view of the system.

CHAPTER III

EXPERIMENTAL RESULTS

Results of applying the procedure outlined in Chapter II to three distinct types of problems shall be presented. The first problem will be a symmetric problem currently existing in the literature. This is done to provide insight to the success of the suppression modification when applied to the Clarke and Wright procedure. The second type problem has an artificially generated cost matrix. The matrix is intended to simulate travel time distortions caused by rush hour traffic problems. The purpose of this problem is to show results of realistic non-symmetry in a cost matrix. The third type of problem has a randomly generated non-symmetric cost matrix. These problems are used to highlight computational aspects of the new procedure.

Symmetric Vehicle Scheduling Problems

Six problems existing in the literature were tested here. Problems 1 and 2 were devised by Hayes to show examples of suboptimal Clarke and Wright solutions. Problem 3 was devised by Tillman and Cochran for the same purpose. Diagrams of the solutions for these three problems are as shown earlier in Figures 12-14. Three additional large problems are tested. These were originally presented by Christofides and Eilon. The results are summarized in Table 6. All solutions shown in the table were obtained from heuristic solution techniques.

Table 6. Comparison of Solution Procedures Used on Symmetric Vehicle Scheduling Problems

Problem	Size, n	Origin
1	7	Hayes
2	6	Hayes
3	7	Tillman & Cochran
4	51	Christofides & Eilon
5	76	" "
6	101	" "

Problem	n	Clarke & Wright[a]			Best Known[b]			Suppression[c]		
		Trks	Time	Cost	Trks	Time	Cost	Trks	Time	Cost
1	7	4		119	2		114	2		114
2	6	3		86	2		75	2		75
3	7	3		420	3		394	3		394
4	51	6	36	591	5	120	546	5	58	573
5	76	10	78	907	10	73	865	10	162	886
6	101	8	150	877	8	240	862	8	124	876

[a] IBM 7090, Gaskell's savings approach.

[b] IBM 360/67, Gillette and Miller's Sweep Algorithm.

[c] Univac 1108.

The suggested modification is successful in determining the critical links of the smaller problems. In each case the optimal solution is obtained. However, improvement of the larger problems is much more difficult to obtain. The additional computation time involved would not normally be justified unless the solution improvement were greater. The worst case possible still gives the Clarke and Wright solution.

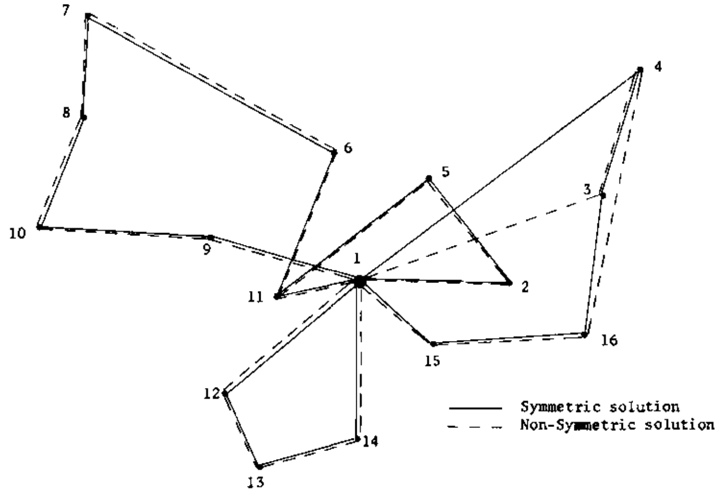
Non-Symmetric Problem Derived from a Symmetric Problem

The second type problem was generated in the following manner. Points were deployed on a Euclidean coordinate system around a central facility, as shown in Figure 25. The Euclidean interpoint distances were determined and a symmetric solution obtained. These Euclidean distances were then perturbed in the following manner. The central facility was assumed to lie at the center of a Central Business District. Travel proceeding along a radial line to or from the center was assumed to be non-symmetric due to rush hour traffic. It would take longer to travel the same distance traveling in heavy traffic than it would traveling in light traffic. More nonsymmetry was assumed to be present the closer travel is to a Central Business District. Travel circumferentially around a Central Business District was assumed symmetric. It would take the same time to travel in either direction. Travel between these two extremes could be either symmetric or non-symmetric. These assumptions were based upon the data collected to generate the non-symmetric cost matrix of Table 1.

This produces an artificial non-symmetric cost matrix as shown in Figure 25. The proposed routes for the non-symmetric case are shown to differ from those of the solution to the symmetric case. In addition, the solution cost is changed, but the difference for this problem is not great. The results indicate that even some assessment of nonsymmetry can affect the solutions.

Random Non-Symmetric Problem

The third type of problem analyzed has a randomly generated



Demand																
	1															
5	9	2														
8	15	8	3													
4	23	17	9	4												
6	7	7	10	17	5											
6	6	12	15	21	5	6										
2	21	28	29	31	21	16	7									
4	18	25	27	31	18	13	5	8								
6	8	17	22	25	12	8	15	10	9							
1	16	25	30	35	20	15	13	8	8	10						
7	4	13	19	24	10	8	20	16	6	13	11					
10	11	19	26	34	18	15	24	19	10	13	8	12				
3	13	20	28	36	20	18	28	23	14	17	11	4	13			
5	8	12	20	26	13	14	28	24	13	20	8	9	8	14		
2	6	4	12	21	7	10	21	23	14	22	9	15	16	8	15	
5	13	4	8	17	11	16	32	29	22	29	17	23	23	14	8	16

Symmetric Cost Matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	--	8	13	10	5	5	18	16	7	14	4	10	11	7	5	12
2	10	--	7	15	7	13	27	24	17	24	14	19	19	12	4	4
3	18	9	--	8	11	17	28	25	23	25	21	27	30	21	14	8
4	25	19	10	--	19	24	30	30	30	36	29	36	38	29	23	16
5	8	8	9	15	--	5	19	17	12	19	11	17	18	13	7	10
6	7	12	13	19	5	--	15	12	8	14	9	14	17	14	10	15
7	24	30	30	31	23	18	--	5	17	14	22	26	30	30	23	34
8	20	26	26	31	19	14	4	--	11	8	18	20	24	25	25	30
9	9	17	21	26	12	7	14	10	--	8	6	10	14	13	14	22
10	18	25	25	34	20	16	13	8	9	--	14	14	17	21	24	30
11	4	13	17	25	10	8	18	15	5	12	--	7	9	8	9	16
12	12	20	26	33	18	16	23	18	11	13	9	--	4	10	16	23
13	14	21	28	35	21	20	27	23	15	17	12	4	--	9	17	23
14	9	12	20	26	14	15	26	23	13	20	9	9	8	--	9	15
15	7	4	12	19	7	9	19	22	13	21	9	14	14	7	--	7
16	15	4	8	17	12	17	31	28	21	29	18	23	23	16	8	--

Related Non-Symmetric Cost Matrix

Figure 25. A Non-Symmetric Problem Derived from a Symmetric Problem

non-symmetric cost matrix. Cost matrices of size 10, 20, 40, 60, 80, and 100 demand points were generated. The solution times for the initial solution procedure (no suppressions) are shown in Figure 26.

Since solution times are very sensitive to problem structure (see Table 7), the results shown in Figure 26 should be taken only as indications. The most important factor, other than the number of demand points, determining solution times appears to be vehicle capacity and the number of points on a route. In Figure 26, problems of the same size have approximately the same number of routes. This should reduce the variability. However, due to this sensitivity and the number of problems solved, the data of Figure 26 should not be taken to show exact relations between symmetric and non-symmetric solution times. The significance is that the non-symmetric problem is no more difficult to solve than the symmetric problem.

Comparing solution times between different researchers can be very misleading. Significant differences can arise from use of different computer systems. Some insight has been given to these problems by Glover, et al. [46]. Of the machines they tested, the slowest machine proved 20 times slower than the fastest. Hence, solution times of different authors on different machines must be compared with this in mind. In addition, specific coding methods and specific problem structure can cause fluctuation in times. This should be considered in any results comparing solution times of different researchers.

Each suppression would require nearly the same time as the initial solution. Initial solution times for some symmetric problems are also shown in the same figure. An interesting result is that the

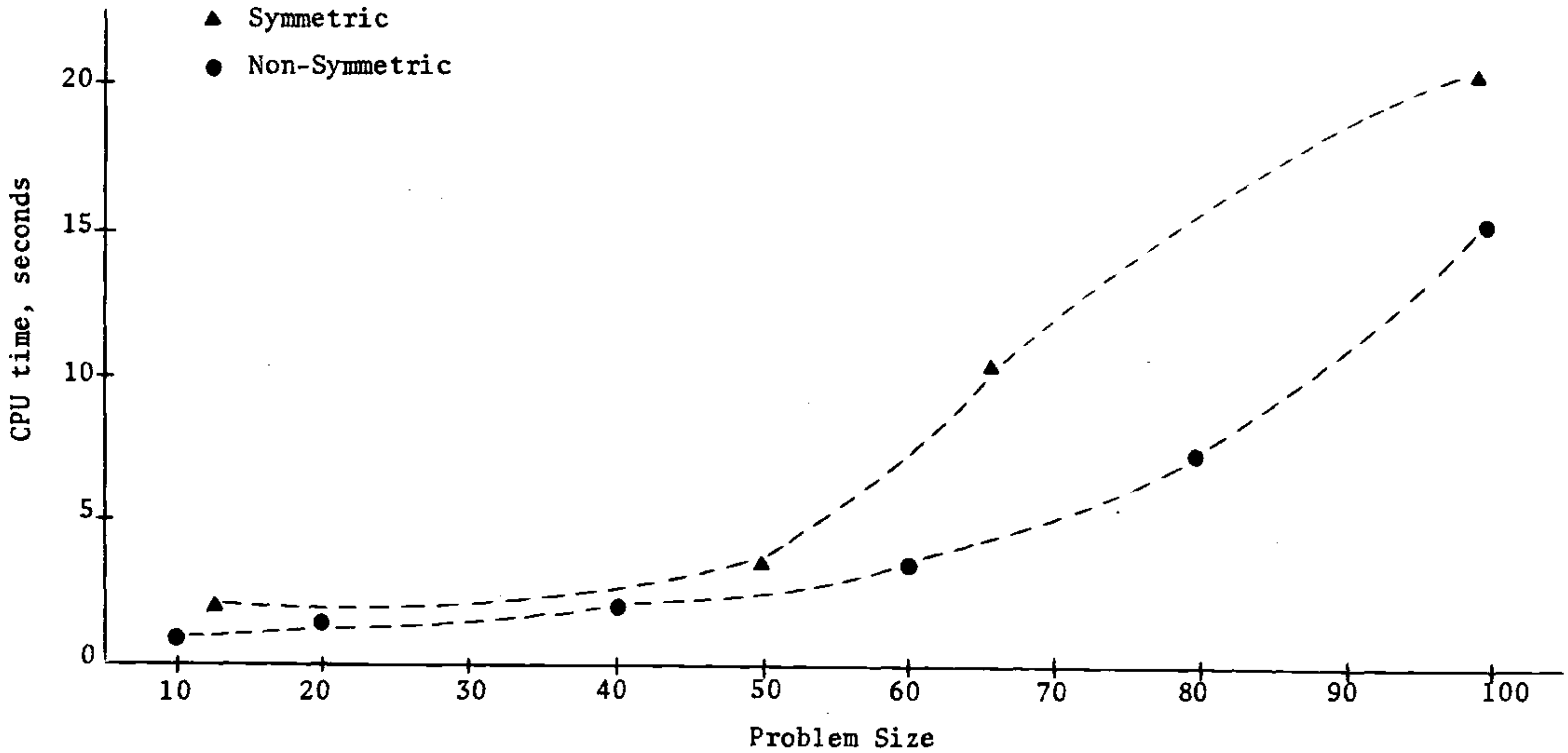


Figure 26. Initial Solution Times for Symmetric and Non-Symmetric Problems

Table 7. Computational Experience with Randomly Generated Non-Symmetric Problems

NON-SYMMETRIC

Vehicle Capacity	Problem Size 20						40						60					
	Initial Solution			Final Solution			Initial Solution			Final Solution			Initial Solution			Final Solution		
	m'	time	K	m'	time	K	m'	time	K	m'	time	K	m'	time	K	m'	time	K
40	6	.45	581	6	5.9	517	12	2.2	884	12	15.6	884	19	9.7	1320	18	68.	1260
60	5	.37	466	4	5.9	432	9	1.6	651	8	22.2	572	14	4.5	971	13	53.	912
80	4	.24	297	4	3.3	297	7	1.15	535	6	12.	511	10	3.3	704	10	54.	604
120																		

Vehicle Capacity	Problem Size 80						100					
	Initial Solution			Final Solution			Initial Solution			Final Solution		
	m'	time	K	m'	time	K	m'	time	K	m'	time	K
40												
60												
80												
120	9	7.2	1014	8	82.	925	11	15.	917	11	193.	857

SYMMETRIC

Vehicle Capacity	Problem Size 51						76						101					
	Initial Solution			Final Solution			Initial Solution			Final Solution			Initial Solution			Final Solution		
	m'	time	K	m'	time	K	m'	time	K	m'	time	K	m'	time	K	m'	time	K
See [9]	6	2.9	591	5	58.	573	10	10.	907	10	162.	886	8	16.5	877	8	124.	876

non-symmetric problems require no more solution time than do the symmetric problems (actually slightly less). This can be attributed to the fact that the size of the matrix to be searched is reduced much more quickly for the non-symmetric problem. Thus despite greater complexity of the initial cost matrix, the non-symmetric case is no more difficult to solve.

The vehicle capacities were varied on the problems of size 20, 40, and 60 demand points. The changes in number of vehicles, total cost, and computational time are summarized in Table 7. In this table both the initial and final solutions are given. The improvements for the random non-symmetric problems are more encouraging than the improvements for the symmetric case. There are two reasons for this. First, because the entries of the non-symmetric cost matrix are completely unrelated, the solutions should show more variation. Secondly, the tendency of the symmetric Euclidean based cost matrix is to produce more related and equivalent solutions. This is a consequence of satisfying the triangle inequality. Bellmore and Malone [7] discuss this in detail.

The results also show that increasing the truck capacities of a problem has three effects: (1) the number of trucks required is reduced; (2) the total solution cost is reduced; and (3) the computation time is reduced. Increasing truck capacities allows more points to be joined which will reduce the total cost. An increase in truck capacity also allows more points to be joined on each route. This requires less routes and less trucks. Since more joinings are feasible, the size of the savings matrix is reduced much more quickly.

The reduced matrix size will require less computational effort and less total time.

The information gained from this type of analysis can be useful when considering the purchase of a new fleet of vehicles or considering additions to a current fleet. The information also specifies what type of problem can be most efficiently solved by this procedure.

CHAPTER IV

CONCLUSIONS AND EXTENSIONS

Conclusions

The algorithm presented extends the treatment of vehicle scheduling problems to include non-symmetric cost matrices. The procedure is capable of solving very large problems (n = 100) without experiencing computation time limitations. In fact, despite the additional complexity of the initial cost matrix, the computation time is no greater than the time needed to solve a symmetric problem of the same size. This observation can be attributed to a much quicker reduction in the size of the associated savings matrix.

It has also been determined that increasing the truck capacity (or alternately, decreasing the number of routes) decreases the computational time necessary to obtain a solution. This decrease, obtained by doubling the vehicle capacity, varied from 47 percent to 63 percent.

A modification to the basic solution procedure has also been introduced. The modification is applicable to both symmetric and non-symmetric problems. The basis for the modification is that some points joined in the original solution may be unfavorable due to truck capacity considerations. Some of the point pairs of the original solution are then "suppressed" to prevent them from appearing in new solutions.

The suppression technique performed well on small symmetric problems which were not solved optimally by the Clarke and Wright procedure. The optimal solution to all of these examples was obtained. The results for large symmetric problems were not particularly favorable. The best solution improvement was only 3 percent. This was obtained with an increase in computation time over that needed for the original solution. In general, the increase in computation time did not justify the solution improvement.

When applied to large non-symmetric problems, the results of suppressing links were much more encouraging. Improvements greater than 14 percent were obtained. The increase in computational effort necessary to achieve this improvement can be seen in Table 7. Because of the quick initial solutions (> 10 seconds, CPU), and the number of solutions generated, this increase in computation time is not critical when justified by the solution improvement. The greater success for non-symmetric problems was attributed to the much greater variety of solutions obtained from the non-symmetric problem. Two additional benefits were derived from the suppression of links: (1) the generation of alternate good quality solutions; (2) the analysis of what parts of the system are most critical.

The algorithm presented here is simple and fast enough to warrant its use when non-symmetric problems are encountered. For these problems it is also recommended that the suppression of links be used.

Extensions

Extensions to this work can be viewed in two areas: (1) extending the procedure itself to solve the types of problems presented here more efficiently; (2) extending the types of solvable problems.

In the first area much improvement can be gained by improving the selection of links to be suppressed. Dominance characteristics of certain links appearing in the solution should exist. Criteria, such as savings/demand, should be investigated as a means of selecting the links to be suppressed. Another criteria could be selecting links for suppression on routes with the most excess capacity. Preliminary attempts to determine more efficient means of selecting links to be suppressed have not been successful. However, criteria such as those mentioned above, should lead to more efficient use of the suppression technique.

The representation of this procedure as a solution tree suggests a branch and bound technique similar to that presented by Bellmore and Malone. Instead of pursuing each branch to termination, a lower bound would be determined for all branches. Only those branches with a lower bound lower than the current best solution would be examined further.

Efforts thus far to find a lower bounding procedure have encountered one of two difficulties. One difficulty is that the bound is not tight enough. Tightness refers to determining a lower bound close enough to the optimal to eliminate a large number of branches from consideration. The second difficulty encountered is that procedures which produce a tight lower bound become as complex as the heuristic solution procedure. There is then no advantage in using a branch and bound technique. However, the efforts to find a bounding technique have been limited. It is felt much improvement could be gained by more work in this area.

Due to a lack of work done on non-symmetric problems, it is difficult to ascertain how close the solutions obtained are to optimality. A method of determining a tight lower bound would also be useful for this purpose. The lower bound could also be used as a criteria for termination. One would terminate when solutions within a desired limit of the lower bound are achieved.

To extend the types of solvable problems, constraints, of the type mentioned at the end of Chapter II, should be considered. An example of such a constraint would be the inclusion of deadlines, i.e., some demands must be serviced within specified time limits. However, as mentioned in Chapter II, the benefits of this additional analysis should be weighed against the cost necessary to achieve it.

The solution procedure presented here may also be extended to problems other than the vehicle scheduling problem. One type of problem which may benefit from this type of analysis would be the problem of performing n jobs on m machines. In this problem, the jobs and machines could be analogous to the demand points and vehicles of the vehicle scheduling problem. Many scheduling problems exhibiting non-symmetric cost matrices could be fertile areas for extension of this solution procedure.

APPENDIX

FORTRAN PROGRAM GUIDE--NOMENCLATURE, SAMPLE INPUT, LISTING, AND SAMPLE OUTPUT

NOMENCLATURE

TERM	Usage
CAPCTY(L)	Represents the capacity of the Lth size truck
DEMAND (I)	Represents the demand to be serviced at the Ith demand point
DIST	Defines a three-dimensional array
DIST (I,J,1)	Represents the cost matrix
DIST (I,J,2)	Represents the savings matrix
DIST (I,J,3)	Represents the current best solution
DIST (I,J,4)	Represents a work area for the solution matrices when suppression is being used
I	Represents the Ith row in the cost or savings matrix
II	Represents the IIth route formed (a route contains more than one demand point)
III	Specifies the order within the file RANK
ICAPCK	Represents an area used to compare the current total demand of a route and vehicle capacities
ICOST	Represents the current best solution cost
ISAVC	Saves the location of a point found on a route
ISAVR	Saves the route number of a point found on a route
ISIZE	Represents a modified SIZE used in DO loops, WRITE statements, etc.
I4	Is a counter specifying the level of suppression
J	Represents a column in the cost or savings matrix
JJ	Represents a position on a route
JCOST	Represents the initial solution cost of one vehicle to each demand point
JOINED	Is used to store elements that currently exist on routes

JOINC	Specifies the column of the point pair picked for joining
JOINR	Specifies the row of the point pair picked for joining
JSAVC	Specifies the location of a point on a route
JSAVR	Specifies the location of a point on a route
K	Specifies which DIST matrix one is using
KK	=1 for the first solution or the current best solution =2 for the solutions of suppressions
K3	=1 for the first solution or the current best solution =2 for the solutions of suppressions
MAXSAV	Represents the current maximum feasible savings
N,N1	Represent the row and column of the current element being suppressed
NEWCST	Represents the current cost of the solution being determined in this iteration
NUMTRK(L)	Represents the number of trucks available of the Lth size
RANK	Represents a file to save the order in which point pairs were joined.
ROUTE	Represents the routes formulated in the solution process
SAVTRK	Is used to save the number of trucks available for re-initializing after each suppression
SIZE	Represents the size of the problem (the number of demand points + the central facility)
SUPP	=0, if suppression is not desired =1, if suppression is desired
SYM	=0, if the problem is non-symmetric =1, if the problem is symmetric
TOUT	Specifies the current amount of CPU time used by the program

```
THE DATA INPUT FOR THIS PROBLEM CONSISTS OF THE FOLLOWING:
THE PROBLEM CONTAINS  69 DEMAND POINTS INCLUDING THE TERMINAL
THE DEMAND AT EACH OF THE POINTS, N=2,....,N, IS:
 0 10 20  5  2 12  7 10  2  4 13  9 18 20 11  6 15  7  5 12
 3  0 17 14  3  9 13 11  4 10  6  2  8 16  4 19 15  7  8 18
10 12  8  6 19 11  7  8 16 10  4  9 13 17  9 12 10 11  6  8
14  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0

THE SIZE AND NUMBER OF TRUCKS AVAILABLE ARE:
TRUCK SIZE              60     0     0     0     0
NUMBER AVAILABLE        99    -1     0     0     0
```

```
C
C
C    THIS MODULE IS DESIGNED AS A HEURISTIC SOLUTION TO THE
C    VEHICLE SCHEDULING PROBLEM.  THE MODULE SOLVES NONSYMMETRIC
C    PROBLEMS HAVING UP TO 101 NODES.  THE COST MATRIX IS
C                        RANDOMLY GENERATED
C
      DIMENSION DIST(101,101,4),DEMAND(101),CAPCTY(5),NUMTRK(5)
      DIMENSION ROUTE(40,23,2),RANK(100,2,2),SAVTRK(5),X(101)
      INTEGER SYM,SUPP,SIZE,DIST,DEMAND,CAPCTY,RANK,ROUTE,SAVTRK
C
C    READ IN THE SIZE AND SET UP THE COST MATRIX
C
      READ(5,1305)(NUMTRK(J),J=1,5),SYM,SUPP,SIZE
      READ(5,1302)(DEMAND(J),J=2,101)
      READ(5,1307)(CAPCTY(J),J=1,5)
      WRITE(6,1319)
      WRITE(6,1320)
      WRITE(6,1321)SIZE
      WRITE(6,1322)
      WRITE(6,1308)(DEMAND(J),J=1,101)
      WRITE(6,1323)
      WRITE(6,1324)(CAPCTY(J),J=1,5)
      WRITE(6,1325)(NUMTRK(J),J=1,5)
      WRITE(6,1326)
C
C   FOLLOWING GENERATES A SIZE X SIZE RAND MATRIX (0-99)
C
57    L=0
      I=1
      READ(5,1310)X(I)
      DO 59 I=1,SIZE
      X(1)=X(I)*100.
      CALL RANDU(X,101)
      DO 58 J=1,SIZE
58    DIST(I,J,1)=X(J)*100.+1.
59    WRITE(6,1308)(DIST(I,JCOL,1),JCOL=1,101)
      DO 60 I=1,5
60    SAVTRK(I)=NUMTRK(I)
      ICOST=0
C
C   INITIALIZE THE SAVINGS TO ONE TRUCK FOR EACH NODE
C
100   DO 110 I=2,SIZE
      ICOST=ICOST+DIST(1,1,1)+DIST(1,I,1)
      DO 110 K=2,4
      DIST(I,1,K)=-1
110   DIST(1,I,K)=-1
      JCOST=ICOST
```

```
C
C     CALCULATE ALL SAVINGS AND STORE THEM IN THREE SEPERATE
C     MATRICES:  ONE FOR THE INITIAL SOLUTION, ONE FOR THE
C     WORK AREA, AND ONE FOR THE CURRENT BEST SOLUTION.
C
      DO 120 I=2,SIZE
      DO 120 J=2,SIZE
      IF(I.EQ.J)GO TO 120
      DIST(I,J,2)=DIST(I,1,1)+DIST(1,J,1)-DIST(I,J,1)
      DIST(I,J,2)=MAX0(DIST(I,J,2),0)
      DIST(I,J,3)=DIST(I,J,2)
120   DIST(I,J,4)=DIST(I,J,2)
C
C     CHECK THE RESULTS SO FAR
C
      WRITE(6,1727)
C     WRITE(6,1308)((DIST(I,J,2),J=1,101),I=1,SIZE)
      WRITE(6,1328)JCOST
C
C     START ASSIGNING ROUTES
C
      I4=1
      K=3
      KK=1
      K3=1
127   NEWCST=JCOST
C     DETERMINE THE CURRENT CPU TIME AND PRINT IT
      IT1=ITIME(IT2,IT3)
      TOUT=IT2
      TOUT=TOUT/5300.
      WRITE(6,1334)TOUT
      III=1
128   JOINR=0
      JOINC=0
      MAXSAV=0
      I=2
C
C     DETERMINE THE MAXIMUM FEASIBLE SAVINGS (MAXSAV)
C
129   J=1
      IF(DIST(I,J,K).EQ.0)GO TO 160
130   J=J+1
      IF(DIST(1,J,K).EQ.0)GO TO 150
      IF(DIST(I,1,K).EQ.0)GO TO 160
      IF(DIST(I,J,K).LE.MAXSAV)GO TO 150
140   JOINR=I
      JOINC=J
      MAXSAV=DIST(I,J,K)
150   IF(J.LT.SIZE)GO TO 130
160   IF(I.EQ.SIZE)GO TO 180
170   I=I+1
      GO TO 129
```

```
C
C CHECK ROUTES FOR FEASIBILITY AND JOIN
C
180    IF(MAXSAV.EQ.0)GO TO 2000
       IF(MAXSAV.LT.0)GO TO 1900
       I=JOINR
       J=JOINC
C    SEE IF BOTH POINTS HAVE ALREADY BEEN ASSIGNED TO A ROUTE
       IF(DIST(J,1,K).EQ.0.AND.DIST(1,I,K).EQ.0)GO TO 600
185    II=1
       JJ=1
       L=2
C
C    CHECK THRU THE EXISTING ROUTES TO EITHER MATCH ONE OF THE
C    POINTS OR ASCERTAIN THAT NO MATCH EXISTS
C
190    JOINED=ROUTE(II,JJ,KK)
       IF(ROUTE(II,23,KK).NE.0)GO TO 205
       IF(JOINED.EQ.0)GO TO 200
       IF(JOINED.EQ.JOINR)GO TO 210
       IF(JOINED.EQ.JOINC)GO TO 250
       JJ=JJ+1
       GO TO 190
200    IF(JJ.EQ.1)GO TO 500
205    JJ=1
       II=II+1
       IF(II.GT.20)GO TO 1900
       GO TO 190
C
C    CHECK CAPACITY REQUIREMENTS AND JOIN IF FEASIBLE
C
210    ICAPCK=ROUTE(II,21,KK)+DEMAND(J)
215    IF(ICAPCK.LE.ROUTE(II,22,KK))GO TO 240
       IF(NUMTRK(2).EQ.-1)GO TO 260
       IF(ICAPCK.LE.CAPCTY(L))GO TO 230
220    L=L+1
       IF(L.EQ.5)GO TO 260
       IF(ICAPCK.GT.CAPCTY(L))GO TO 220
230    IF(NUMTRK(L).EQ.0)GO TO 220
       ROUTE(II,22,KK)=CAPCTY(L)
       NUMTRK(L)=NUMTRK(L)-1
       IF(JOINED.EQ.0.AND.JJ.EQ.1)GO TO 240
       NUMTRK(L-1)=NUMTRK(L-1)+1
240    ROUTE(II,21,KK)=ICAPCK
       IF(ROUTE(II,JJ+1,KK).NE.0)GO TO 310
       NEWCST=NEWCST-DIST(I,J,2)
       ROUTE(II,JJ+1,KK)=JOINC
       IF(JOINED.EQ.0.AND.JJ.EQ.1)ROUTE(II,JJ,KK)=JOINR
```

```
C
C    UPDATE THE SAVINGS MATRIX TO REFLECT THE NEW JOINED PAIR
C
243     DIST(I,1,K)=0
        DIST(1,J,K)=0
1307    FORMAT(5I6)
C    SAVE THE ORDER IN WHICH THIS PAIR WAS JOINED
        RANK(III,1,K3)=I
        RANK(III,2,K3)=J
        III=III+1
        IF(III.GT.100)GO TO 1900
245     DIST(I,J,K)=-1
        DIST(J,I,K)=0
        GO TO 128
250     JOINC=JOINR
        ICAPCK=ROUTE(II,21,KK)+DEMAND(I)
        GO TO 215
260     DIST(I,J,K)=0
        GO TO 128
C
C    NO MATCH HAS BEEN ON AN EXISTING ROUTE.  JOIN THESE POINTS
C    ON A NEW ROUTE
C
300     L=1
        ICAPCK=DEMAND(I)+DEMAND(J)
        IF(CAPCTY(L).LT.ICAPCK)GO TO 220
        NUMTRK(L)=NUMTRK(L)-1
        ROUTE(II,21,KK)=ICAPCK
        ROUTE(II,22,KK)=CAPCTY(L)
270     ROUTE(II,JJ,KK)=JOINR
        ROUTE(II,JJ+1,KK)=JOINC
        NEWCST=NEWCST-DIST(I,J,K)
        GO TO 243
C
C THE NEW POINT ON THIS ROUTE MUST BE INSERTED AT THE BEGINNING
C    SHIFT THE OTHER POINTS TO ALLOW FOR THIS
C
310     IF(JJ.NE.1)GO TO 1900
315     JJ=JJ+1
        IF(JJ.EQ.19)GO TO 1900
        IF(ROUTE(II,JJ+1,KK).NE.0)GO TO 315
320     ROUTE(II,JJ+1,KK)=ROUTE(II,JJ,KK)
        IF(JJ.EQ.1)GO TO 330
        JJ=JJ-1
        GO TO 320
330     ROUTE(II,JJ,KK)=JOINC
        NEWCST=NEWCST-DIST(I,J,K)
        GO TO 243
C
C    AN ERROR HAS OCCURED.  PRINT SOME HELPFUL INFO AND TERM
C
```

```
1900   WRITE(6,1301)I,J,K,II,JJ,KK,K3,MAXSAV,ICOST,ICAPCK
       GO TO 2100
C
C    A SOLUTION HAS BEEN REACHED.   NOTIFY AND CONTINUE BY
C    CHECKING FOR SUPPRESSION
C
2000   IF(SUPP.EQ.0)GO TO 2100
       IF(K.NE.3)I4=I4+1
       IP4=I4-1
C
C    OBTAIN THE NEXT ELEMENT OF THE ORDERED JOINED PAIRS(RANK)
C
C    IF INITIAL SOLUTION RE-INITIALIZE VALUES AND RETURN
C
       N=RANK(I4,1,1)
       N1=RANK(I4,2,1)
       IF(K.NE.3)GO TO 360
       WRITE(6,1328)NEWCST
       DO 350 I=1,5
350    NUMTRK(I)=SAVTRK(I)
       DIST(N,N1,4)=0
       K=4
       KK=2
       K3=2
       ICOST=NEWCST
       GO TO 127
C
C    SEE IF THE NEW SOLUTION IS BETTER THAN THE CURRENT BEST
C    IN EITHER CASE SAVE THE BEST SOLUTION, RE-INITIALIZE THE
C    PROPER VALUES AND RETURN FOR THE NEXT ITERATION
C
360    IF(ICOST.GT.NEWCST)GO TO 380
       WRITE(6,1329)IP4,NEWCST
377    CONTINUE
       IF(RANK(I4,1,1).EQ.0)GO TO 2100
       IF(I4.EQ.6)GO TO 2100
       DO 370 M1=1,SIZE
       DO 370 M2=1,SIZE
370    DIST(M1,M2,4)=DIST(M1,M2,2)
       DIST(N,N1,4)=0
       DO 375 I=1,5
375    NUMTRK(I)=SAVTRK(I)
       DO 378 I=1,40
       DO 378 J=1,23
378    ROUTE(I,J,2)=0
       GO TO 127
380    CONTINUE
       WRITE(6,1330)IP4
       WRITE(6,1331)NEWCST
       DO 390 I=1,SIZE
       DO 390 J=1,SIZE
```

```
390    DIST(M1,M2,3)=DIST(M1,M2,4)
       ICOST=NEWCST
       DO 400 M3=1,40
       DO 400 M4=1,23
400    ROUTE(M3,M4,1)=ROUTE(M3,M4,2)
       ROUTE(M3,M4,2)=0
       N=RANK(I4-1,1,1)
       N1=RANK(I4-1,2,1)
       DIST(N,N1,2)=0
       I4=1
       DO 410 M4=1,2
       DO 410 M3=1,80
       RANK(M3,M4,1)=RANK(M3,M4,2)
410    RANK(M3,M4,2)=0
       N=RANK(I4,1,1)
       N1=RANK(I4,2,1)
       GO TO 377
C
C BOTH POINTS OF MAXSAV ARE CURRENTLY ON A ROUTE.  FIND
C    THE ROUTES, JOIN THEM IN THE PROPER MANNER, AND UPDATE THE
C    DATA
C
600    ISAVR=0
       ISAVC=0
       JSAVR=0
       JSAVC=0
       II=1
610    JJ=1
       IF(ROUTE(II,23,KK).NE.0)GO TO 620
615    IF(ROUTE(II,JJ,KK).LE.0)GO TO 620
       IF(I.EQ.ROUTE(II,JJ,KK))GO TO 630
       IF(J.EQ.ROUTE(II,JJ,KK))GO TO 640
617    JJ=JJ+1
       IF(JJ.GT.20)GO TO 1900
       GO TO 615
620    IF(II.EQ.40)GO TO 1900
       II=II+1
       GO TO 610
630    ISAVR=II
       ISAVC=JJ
       IF(JSAVR.GT.0)GO TO 650
       GO TO 617
640    JSAVR=II
       JSAVC=JJ
       IF(ISAVR.GT.0)GO TO 650
       GO TO 617
650    L=1
       IF(JSAVR.EQ.ISAVR)GO TO 200
       ICAPCK=ROUTE(ISAVR,21,KK)+ROUTE(JSAVR,21,KK)
660    IF(ICAPCK.LT.CAPCTY(L))GO TO 670
```

```
      IF(L.EQ.5)GO TO 260
      L=L+1
      GO TO 660
670   IF(ISAVC.EQ.1)GO TO 700
      IF(JSAVC.NE.1)GO TO 740
      JJ=2
680   IF(ROUTE(JSAVR,JJ+1,KK).EQ.0)GO TO 690
      JJ=JJ+1
      GO TO 680
690   ROUTE(ISAVR,ISAVC+1,KK)=ROUTE(JSAVR,JJ,KK)
      GO TO 750
700   IF(JSAVC.EQ.1)GO TO 710
      ROUTE(ISAVR,1,KK)=ROUTE(JSAVR,1,KK)
      GO TO 750
710   JJ=2
720   IF(ROUTE(JSAVR,JJ+1,KK).EQ.0)GO TO 730
      JJ=JJ+1
      GO TO 720
730   ROUTE(ISAVR,1,KK)=ROUTE(JSAVR,JJ,KK)
      GO TO 750
740   ROUTE(ISAVR,ISAVC+1,KK)=ROUTE(JSAVR,1,KK)
750   ROUTE(JSAVR,23,KK)=ISAVR
      ROUTE(ISAVR,21,KK)=ICAPCK
      ROUTE(ISAVR,22,KK)=CAPCTY(L)
      NUMTRK(L)=NUMTRK(L)-1
      L=1
760   IF(CAPCTY(L).EQ.ROUTE(JSAVR,22,KK))GO TO 770
      IF(L.EQ.5)GO TO 1900
      L=L+1
      GO TO 760
770   NUMTRK(L)=NUMTRK(L)+1
      NEWCST=NEWCST-DIST(I,J,2)
      GO TO 243
C
C   THE ENTIRE SOLUTION PROCEDURE IS FINISHED.  PRINT THE
C   BEST SOLUTION WITH BACKUP INFO AND TERMINATE
C
2100  WRITE(6,1332)ICOST
      WRITE(6,1333)
      DO 500 I=1,40
500   WRITE(6,1303)(ROUTE(I,J,1),J=1,23)
1304  FORMAT(I5)
      WRITE(6,1307)(NUMTRK(I),I=1,5)
1301  FORMAT(10I4)
1309  FORMAT(20I4,3I15)
1306  FORMAT(20I3/20I3/20I3/20I3/20I3/I3)
1310  FORMAT(F10.5)
1303  FORMAT(20I4/20I4/20I4/20I4/20I4/I4)
1302  FORMAT(20I4)
1305  FORMAT(5I6/3I3)
```

```
1318  FORMAT(6I5)
1319  FORMAT(1H1,14X,'---OUTPUT FOR A NON-SYMMETRIC VEHICLE
     &PROBLEM---')
1320  FORMAT(1H0,'THE DATA INPUT FOR THIS PROBLEM CONSISTS OF
     &TWO:')
1321  FORMAT(' THE PROBLEM CONTAINS ',I3,' DEMAND POINTS
     &TERMINAL')
1322  FORMAT(' THE DEMAND AT EACH OF THE POINTS, N=2,...,N, IS:'
1323  FORMAT(//' THE SIZE AND NUMBER OF TRUCKS AVAILABLE ARE:')
1324  FORMAT(' TRUCK SIZE ',7X,5I5)
1325  FORMAT(' NUMBER AVAILABLE   ',5I5)
1326  FORMAT(1H1,'THE FOLLOWING COST MATRIX HAS BEEN USED:')
1327  FORMAT(1H1,'FROM THE ABOVE COST MATRIX THE FOLLOWING
     &X HAS BEEN CALCULATED:')
1328  FORMAT(1H1,'THE COST OF THE FIRST SOLUTION IS ',I7)
1329  FORMAT(1H0,'SUPPRESSION NUMBER ',I2,' OF THIS SOLUTION
     &T OF ',I7)
1330  FORMAT(1H0,'SUPPRESSION NUMBER ',I2,' HAS IMPROVED',
     &' THE SOLUTION')
1331  FORMAT(1H0,'THE NEW CURRENT BEST SOLUTION',
     &' HAS A COST OF ',I7)
1332  FORMAT(1H1,'THE FINAL SOLUTION HAS A COST OF ',I7,///' THE
     &USED IN THIS SOLUTION ARE AS SHOWN BELOW')
1333  FORMAT(1H0,'ROUTES',T35,'DEMAND',T100,'CAP',
     &'ACITY',T115,'ACTIVITY')
1334  FORMAT(1H0,'  THE TOTAL CPU TIME USED THUS FAR IS ',F9.5)
      STOP
      END
```

THE COST OF THE FIRST SOLUTION IS 805

THE TOTAL CPU TIME USED THUS FAR IS 5.24240

SUPPRESSION NUMBER 1 HAS IMPROVED THE SOLUTION

THE NEW CURRENT BEST SOLUTION HAS A COST OF 784

THE TOTAL CPU TIME USED THUS FAR IS 9.45240

SUPPRESSION NUMBER 1 OF THIS SOLUTION GIVES A COST OF 785

THE TOTAL CPU TIME USED THUS FAR IS 13.28900

SUPPRESSION NUMBER 2 HAS IMPROVED THE SOLUTION

THE NEW CURRENT BEST SOLUTION HAS A COST OF 761

THE TOTAL CPU TIME USED THUS FAR IS 17.05440

SUPPRESSION NUMBER 1 HAS IMPROVED THE SOLUTION

THE NEW CURRENT BEST SOLUTION HAS A COST OF 744

THE TOTAL CPU TIME USED THUS FAR IS 20.92780

SUPPRESSION NUMBER 1 OF THIS SOLUTION GIVES A COST OF 779

THE TOTAL CPU TIME USED THUS FAR IS 25.07520

SUPPRESSION NUMBER 2 OF THIS SOLUTION GIVES A COST OF 796

THE TOTAL CPU TIME USED THUS FAR IS 29.17340

SUPPRESSION NUMBER 3 OF THIS SOLUTION GIVES A COST OF 792

THE TOTAL CPU TIME USED THUS FAR IS 32.98620

SUPPRESSION NUMBER 4 OF THIS SOLUTION GIVES A COST OF 832

THE TOTAL CPU TIME USED THUS FAR IS 37.03220

SUPPRESSION NUMBER 5 OF THIS SOLUTION GIVES A COST OF 782

THE FINAL SOLUTION HAS A COST OF 744

THE ROUTES USED IN THIS SOLUTION ARE AS SHOWN BELOW

ROUTES						DEMAND	CAPACITY	ACTIVITY
42	6	33	51	37	31	57	60	0
17	29	55	21	0	0	43	60	0
56	55	0	0	0	0	21	60	2
57	49	8	0	0	0	46	60	0
32	14	0	0	0	0	22	60	10
43	45	15	54	0	0	50	60	0
46	2	60	25	0	0	43	60	0
58	30	27	35	52	0	52	60	0
16	20	38	23	0	0	42	60	0
28	41	14	53	0	0	56	60	0
36	47	12	7	18	0	57	60	0
11	22	44	24	0	0	41	60	0
13	10	4	3	0	0	56	60	0
19	25	0	0	0	0	8	60	7
26	3	0	0	0	0	29	60	13
34	5	39	0	0	0	32	60	0
59	9	40	0	0	0	26	60	0
48	18	0	0	0	0	15	60	11
50	8	0	0	0	0	20	60	4
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

REFERENCES

1. "Annual Report of the Postmaster General," 1972-1973.

2. Balinski, M. L. and R. E. Quandt, "On an Integer Program for a Delivery Problem," Operations Research, Vol. 12, No. 2, 1964.

3. Bazaraa, M. S. and J. J. Goode, "The Traveling Salesman Problem: A Duality Approach," Georgia Institute of Technology, 1974.

4. Bellman, R., "On a Routing Problem," Quarterly Journal of Applied Mathematics, Vol. 26, 1958.

5. Bellmore, M. and G. L. Nemhauser, "The Traveling Salesman Problem: A Survey," Operations Research, Vol. 16, 1968.

6. Bellmore, M., G. Bennington, and S. Lufore, "A Multivehicle Tanker Scheduling," Transportation Science, Vol. 3, 1970.

7. Bellmore, M. and J. Malone, "Pathology of Traveling-Salesman Subtour-Elimination Algorithms," Operations Research, Vol. 19, 1971.

8. Beltrami, E. J. and L. D. Bodin, "Networks and Vehicle Routing for Municipal Waste Collection," Networks, Vol. 4, 1974.

9. Christofides, N. and S. Eilon, "An Algorithm for the Vehicle Dispatching Problem," Operational Research Quarterly, Vol. 20, 1969.

10. Clarke, G. and J. W. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," Operations Research, Vol. 12, 1964.

11. Cochran, H., "Optimization of a Carrier Routing Problem," Master's Thesis, Kansas State University, Manhatten, Kansas, 1968.

12. Dantzig, G. B., D. R. Fulkerson, and S. M. Johnson, "Solution of a Large-Scale Traveling Salesman Problem," Operations Research, Vol. 2, 1954.

13. Dantzig, G. B. and Ramser, J. H., "The Truck Dispatching Problem," Management Science, Vol. 6, No. 1, 1959.

14. Dun, R. A., "A Two-Phase Algorithm for the Vehicle Delivery Problem," M.S. Thesis, Georgia Institute of Technology, Atlanta, Georgia, 1971.

15. Edmonds, J. and E. L. Johnson, "Matching, Euler Tours, and the Chinese Postman Problem," Mathematical Programming, Vol. 5, No. 1, 1973.

16. Encyclopedia Brittannica, 1973.

17. Flood, M., "The Traveling-Salesman Problem," Operations Research, Vol. 4, 1956.

18. Ford, L. R. and D. R. Fulkerson, Flows in Networks, Princeton University Press, 1962.

19. Gaskell, T. J., "Bases for Vehicle Fleet Scheduling," Operational Research Quarterly, Vol. 18, No. 3, 1967.

20. Garvin, W. W., H. W. Crandell, J. B. John, and R. A. Spellman, "Application of Linear Programming in the Oil Industry," Management Science, Vol. 3, 1957.

21. Gillette, B. E., and L. R. Miller, "A Heuristic Algorithm for the Vehicle-Dispatch Problem," Operations Research, Vol. 22, No. 2, 1974.

22. Gomory, R. E., "Outline of an Algorithm for Integer Solutions of Linear Problems," Bulletin of the American Mathematical Society, 64, 1958.

23. Hayes, R. L., "The Delivery Problem," Master's Thesis, Carnegie Institute of Technology, 1967.

24. Held, M. and R. M. Karp, "The Traveling Salesman Problem and Minimum Spanning Trees," Operations Research, Vol. 18, No. 6, 1970.

25. Hering, R., "Evaluation of Heuristic Look-Ahead Rules for the Delivery Problem," M.S. Thesis, Kansas State University, Manhatten, Kansas, 1970.

26. Karg, R. L. and G. L. Thompson, "A Heuristic Approach to Solving Traveling Salesman Problems," Management Science, No. 10, 1964.

27. Levin, A., "Scheduling and Fleet Routing Models for Transportation Systems," Transportation Science, Vol. 5, 1971.

28. Lin, S., "Computer Solutions of the Traveling Salesman Problem," Bell System Technical Journal, Vol. 44, 1965.

29. Lin, S. and B. W. Kernighan, "An Effective Heuristic Algorithm for the Traveling Salesman Problem," Operations Research, Vol. 21, 1973.

30. Little, J. D. C., K. G. Murty, D. W. Sweeney, and C. Karel, "An Algorithm for the Traveling Salesman Problem," Operations Research, Vol. 4, No. 6, 1963.

31. Martin-Loef, A., "A Branch and Bound Algorithm for Determining the Minimal Fleet Size of a Transportation System," Transportation Science, Vol. 4, 1970.

32. McDonald, J. J., "Vehicle Scheduling: A Case Study," Operational Research Quarterly, Vol. 23, No. 4, 1972.

33. Newton, K. and W. Thomas, "Developing a Computer Program for Bus Routing: Final Report," State University of New York Research Foundation, Albany, New York, July 1970.

34. O'Neill, B. F. and D. C. Whyback, "Vehicle Routing from Central Facilities," International Journal of Physical Distribution Management, February, 1972.

35. Orloff, C. S., "A Fundamental Problem in Vehicle Routing," Networks, Vol. 4, 1974.

36. Orloff, C. S., "Routing a Fleet of M Vehicles to/from a Central Facility," Networks, Vol. 4, 1974.

37. Pierce, J. F., "Direct Search Algorithms for Truck-Dispatching Problems," Transportation Research, Vol. 3, No. 1, 1969.

38. Pierce, J. F., "On the Truck Dispatching Problem: Part I," IBM Cambridge Scientific Center Report 320-3218, Cambridge, Massachusetts, 1967.

39. Smith, A., "A Heuristic Approach to the Line Haul Problem," Master's Thesis, Kansas State University, Manhatten, Kansas, 1971.

40. Svestka, J. and V. Huckfeldt, "Computational Experience with an M-Salesman Traveling Salesman Algorithm," Management Science, Vol. 19, 1973.

41. Tillman, F. A., "The Multiple Terminal Delivery Problem with Probabilistic Demands," Transportation Science, Vol. 3, No. 3, 1969.

42. Tillman, F. A. and H. Cochran, "A Heuristic Approach for Solving the Delivery Problem," Journal of Industrial Engineering, Vol. 19, No. 7, 1968.

43. "Description of Mail Preparation Operations at the Atlanta Post Office," Internal Report, Georgia Institute of Technology, Atlanta, Georgia, 1974.

44. Mueller-Merbach, H., Optimale Reihenfolgen, Springer Verlag, 1970.

45. Ullman, Z., "Optimal Assignment of Postal Vehicles for Local Haul," Research Memorandum ORD-RM 6748-7 for Stanford Research Institute Project 6748, January, 1969.

46. Glover, F., D. Karney, D. Klingman, "Implementation and Computational Comparisons of Primal, Dual, and Primal-Dual Computer Codes for Minimum Cost Network Flow Problems," Networks, Vol. 4, 1974.